



Programmes d'analyse des resultats d'une classification automatique

Israël-César Lerman

► To cite this version:

Israël-César Lerman. Programmes d'analyse des resultats d'une classification automatique. [Rapport de recherche] RR-0168, INRIA. 1982. inria-00076390

HAL Id: inria-00076390

<https://hal.inria.fr/inria-00076390>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Novembre 1982

PROGRAMMES D'ANALYSE DES RÉSULTATS D'UNE CLASSIFICATION AUTOMATIQUE

par I.C. LERMAN et Collaborateurs (*)

RESUME : C'est un ensemble de programmes qui, supposant déjà acquis le résultat d'une "bonne" classification de l'ensemble des lignes (resp. colonnes) d'un tableau de données, gravitent autour de la notion de classe "naturelle" de proximité. Ils permettent, pour différentes structures du tableau des données, d'"expliquer" de façon contrôlée et systématique la raison des classes formées; ils permettent d'autre part, de croiser des typologies, de déterminer des échelles hiérarchiques d'attitude et enfin, de "mesurer" l'aptitude à être classifié d'un ensemble muni d'une préordonnance totale.

Le détail des aspects méthodologiques peut être étudié dans l'ouvrage de I.C. Lerman : "Classification et analyse ordinale des données", Dunod, Paris (1981).

SUMMARY : We propose a collection of independant programs for interpreting and analyzing classification -of a set of units of data- which is generally defined by any given clustering algorithm. These programs correspond to statistical and combinatorial methods building around the concept of "natural" classe of individuals (resp. variables). The study of these methods may be consulted in the book of I.C. Lerman : "Classification et analyse ordinale des données", Dunod, Paris (1981).

The purposes of these programs are :

- . to "explain" in a controlled and systematic way, the reason of the forming of the classes ; that is to say, by what the classes may be characterized,
- . to cross fuzzy typologies,
- . to determine scaling for multidimensionnal ordinal variables,
- . to measure the ability to be classified of a set provided by its "preordonnance" (i.e. total ranking on the set of its unordered pairs).

(*) IRISA, Laboratoire de Statistique, Campus de Beaulieu,
Avenue du Général Leclerc, 35042 RENNES CEDEX.



I - INTRODUCTION GENERALE

Nous avons tenu à regrouper ici l'ensemble de nos programmes qui permettent l'analyse contrôlée des résultats d'une classification automatique des lignes ou des colonnes d'un tableau de données compte tenu de la structure mathématique de ce dernier.

Ces programmes ont été élaborés dans le cadre de type de données réellement rencontrées dans nos recherches de sorte que nous ne traitons pas de façon systématique l'ensemble de tous les types de tableaux de données (tels qu'ils ont pu par exemple être définis dans le chapitre 2 de l'ouvrage "Classification et analyse ordinale des données"). D'autre part, pour un même type de données, nous ne considérons pas de façon symétrique aussi bien les lignes que les colonnes du tableau des données et ce, compte tenu de la problématique du chercheur-utilisateur.

D'où, une certaine hétérogénéité ; une autre raison de cette hétérogénéité est due à ce que les différents programmes ont respectivement été conçus par différents auteurs, que nous mentionnons ci-dessous. Certains de ces programmes (les trois premiers) ont d'ailleurs été élaborés dans le cadre de thèses de 3ème cycle. D'autre part, le degré d'explicitation du contenu algorithmique des différents programmes n'est sans doute pas identique ; de toute façon, nous avons cherché autant que possible à respecter pour chacun des programmes le plan de présentation suivant :

- 1 - Fonction générale du programme
- 2 - Calculs effectués
- 3 - Paramètres et données du programme
- 4 - Organisation et algorithmes
- 5 - Listage.

Malgré cette absence de parfaite homogénéité, nous espérons fournir ici un document suffisamment clair et d'une très grande utilité qui fait directement suite à celui intitulé "Programme de classification hiérarchique" (rapport interne IRISA n° 148, juin 1981).

Nous allons chercher à indiquer rapidement les fonctions générales des différents programmes à travers la suite des paragraphes suivants :

II - DISCRIMINATION DES CLASSES BASEE SUR LA DECOMPOSITION DE LA VARIANCE.

II.1. DEGRE DE RESPONSABILITE DANS LE CAS DE DONNES NUMERIQUES ; par J.P. Geffrault

Relativement à une classification ou à une suite de quelques classifications emboîtées de l'ensemble des individus ou objets, ce programme permet de déterminer, sous la forme d'un rapport de variances, les degrés de responsabilités des différentes variables ou classes de variables ; lesquelles pouvant en l'occurrence être obtenues à partir d'une classification par proximité de l'ensemble des variables de description. L'organisation des calculs est basée sur une gestion de la représentation polonaise de l'arbre des classifications.

II.2. DEGRE DE RESPONSABILITE DANS LE CAS D'UN TABLEAU D'INCIDENCE DES DONNEES ; par T. Chantrel

Relativement à une typologie définie par une classification de l'ensemble des attributs descriptifs (lesquels pouvant d'ailleurs correspondre aux

modalités de variables qualitatives d'un questionnaire), on détermine pour chacun des sujets, le degré de responsabilité global par rapport à la classification des attributs et ceux locaux, relativement à chacune des classes d'attributs. Le degré de responsabilité global permet de préciser le caractère "typique" d'un individu et ceux locaux, le caractère "typique d'un profil d'attitude". D'autre part, par rapport aux classes d'individus respectivement définies par une variable qualitative exogène, on détermine pour chacune de ces classes, son degré d'intervention dans la définition des différents profils d'attitude, en termes de la proportion dans la classe de sujets, de ceux typiques d'un profil de comportement défini par une classe d'attributs.

II.3. DEGRE DE RESPONSABILITE DANS LE CAS DE DONNEES DE CONTINGENCE ; par A. Prod'homme

La donnée est un tableau résultant de la juxtaposition "horizontale" de tables de contingence dont l'ensemble des lignes est indexé par un ensemble I (pouvant par exemple représenter un ensemble d'unités géographiques) et dont l'ensemble des colonnes, par un ensemble H de la forme $H = J^{(1)} \cup J^{(2)} \cup \dots \cup J^{(h)} \cup \dots \cup J^{(k)}$, où $J^{(h)}$, $1 \leq h \leq k$, désigne l'ensemble des modalités d'une variable qualitative.

On suppose établie une classification $\{I_\ell / 1 \leq \ell \leq L\}$ sur I, à partir notamment de sa description à travers H. Pour expliquer ou décrire synthétiquement la classification sur I, on définit et on calcule le degré de responsabilité de chaque modalité j_0 de $J^{(h)}$, $1 \leq h \leq k$, dans la formation de chacune des classes I_ℓ , $1 \leq \ell \leq L$.

III - CROISEMENT DE CLASSIFICATIONS "FLOUES" ; par M. Hardouin

S et C sont deux ensembles disjoints d'attributs de description d'un même ensemble E d'individus ou objets ; S et C sont généralement associés à deux aspects sémantiques différents d'une même étude. On considère deux partitions de S et de C :

$\pi(S) = \{S_r / 1 \leq r \leq h\}$ et $\pi(C) = \{C_s / 1 \leq s \leq k\}$ qui correspondent généralement à deux classifications polythétiques par proximité de S et de C et qui définissent deux typologies de l'ensemble E des individus.

Ce programme établit la table de croisement entre $\pi(S)$ et $\pi(C)$, où à l'intersection de la ligne r et de la colonne s ($1 \leq r \leq h$, $1 \leq s \leq k$), on trouve la valeur d'un indice, conforme à la statistique du χ^2 , mesurant le lien entre le profil d'attitude défini par la classe S_r et celui défini par la classe C_s . Un usage courant de ce programme est celui où chaque classe S_r contient exactement un attribut-modalité d'une variable qualitative.

Signalons également qu'un sous-produit intéressant de ce programme est le calcul de l'importance numérique d'un profil de comportement défini par une classe d'attributs orientés.

IV - ANALYSE D'UNE CLASSE D'ECHELLES ; par C. Riso-Levi

La donnée est une classe de variables qualitatives ordinales ; nous disons encore de caractères descriptifs totaux, où l'ensemble des modalités d'un même caractère est totalement ordonné. On suppose qu'une telle classe

recouvre une même variable sous-jacente par rapport à laquelle on peut totalement ordonner l'ensemble des modalités qui n'occupent pas la position 0 des différents caractères, de façon compatible avec chacun des ordres défini sur l'ensemble des modalités d'un même caractère.

On présente ici, conformément à la méthode exprimée au chapitre 9 (§III) de "Classification et analyse ordinaire des données", un ensemble cohérent de programmes de recherche de l'"échelle hiérarchique" sous-jacente à une classe d'items totaux, permettant d'insérer entre elles les modalités qui n'occupent pas la position initiale des différents items.

On peut s'étonner de voir ce programme de portée générale et concernant un problème classique posé par les psychologues mathématiciens, s'inscrire dans l'ensemble des programmes d'analyse des résultats d'une classification automatique. En effet, de notre point de vue, il est arbitraire de délimiter a priori une classe de variables qualitatives ordinales sensées se référer à une même échelle sous-jacente. Nous obtenons quant à nous une telle classe au moyen d'une classification par proximité d'un vaste ensemble de telles variables. D'autre part, de la sorte, on porte l'accent sur le fait que les algorithmes présentés ici représentent un outil pour l'interprétation des résultats d'une classification automatique d'une famille d'échelles.

V - CLASSIFICABILITE ; par I.C. Lerman

Il s'agit d'un ensemble de programmes qui développent l'étude de la "classificabilité" (cf. chap.3, § IV de "Classification et analyse ordinaire des données"). Ces programmes permettent d'une part, de calculer la distribution $D(\omega)$ qui caractérise le degré de classificabilité d'un ensemble dont la préordonnance totale est ω , dans un cas réel comme dans un cas simulé, dans le cadre de l'hypothèse d'absence de lien N_1 pour des données d'incidence. Ils permettent d'autre part de calculer une telle distribution pour un tableau d'incidence aléatoire défini dans l'h.a.l. N_1 , pour le cas particulier où le nombre de composantes égales à 1 d'un même vecteur logique de description, est constant.

Alors que les autres programmes (paragraphe I à IV) n'interviennent qu'une fois acquis le résultat d'une classification automatique des données; les programmes sur la classificabilité (ou aptitude à être classifié) d'un ensemble, sont à considérer avant toute classification.

Comme nous l'avons déjà signalé ci-dessus, on peut relativement aux thèmes abordés en II et III ci-dessus, créer un système plus complet de programmes, répondant aux différentes structures possibles d'un tableau de données et travaillant symétriquement aussi bien par rapport aux lignes qu'aux colonnes du tableau des données. La mise au point d'un tel logiciel suppose bien entendu de devoir préciser certains aspects statistiques liés à l'élaboration des coefficients à mettre en oeuvre. Nous allons proposer, à titre d'exemple, quelques situations complémentaires de certaines déjà envisagées ci-dessus.

Dans le cas d'un tableau de données numériques, on peut considérer une situation duale de celle étudiée par J.P. Geffrault et chercher, relativement à une classification par proximité de l'ensemble des variables, à associer à chaque classe de variables, au moyen d'un indice, les individus qui justifient le mieux une telle classe. D'autre part, on peut chercher à situer par rapport à cette classe de variables les différentes classes

d'individus définies par une variable exogène qualitative. Une telle recherche serait le pendant de celle effectuée par T. Chantrel dans le cas où les variables sont des attributs de description.

De façon symétrique ; c'est-à-dire duale de celle envisagée par T. Chantrel dans le cas des tableaux d'incidence, on peut demander à avoir un traitement analogue à celui effectué par J.P. Geffrault dans le cas des données numériques. En d'autres termes, relativement à une même classe d'individus (d'une classification par proximité de l'ensemble des individus) déterminer le degré de discrimination de cette classe par un attribut donné ou une classe d'attributs et circonscrire les attributs les plus responsables de la formation de cette classe.

Ce type d'étude où, répétons-le, relativement à deux classifications par proximité de l'ensemble des variables descriptives d'une part et de l'ensemble des individus d'autre part, on détermine le degré de responsabilité d'un individu ou classe d'individus (resp. d'une variable ou classe de variables) dans la formation d'une classe de variables (resp. d'individus) ; peut également être envisagé dans le cas où les variables sont qualitatives ordinales.

Enfin, on doit pouvoir étendre les formules et la technique du croisement entre classifications "floues" à d'autres structures de données ; notamment et surtout au cas où les données sont quantitatives numériques.

L'idéal est de pouvoir intégrer les calculs des paragraphes II et III et les extensions que nous venons de mentionner dans le cadre d'un même logiciel dont le point de départ peut être le programme de J.P. Geffrault qui se distingue par la manière dont il gère l'information.

(*) "Programme de classification hiérarchique", I.C. Lerman, Publication interne IRISA, n° 148, Juin 1981.

II - DISCRIMINATION DES CLASSES BASEE SUR LA DECOMPOSITION DE LA VARIANCE

II.1. DEGRE DE RESPONSABILITE DANS LE CAS DE DONNEES NUMERIQUES

PROGRAMME RESPNUM (J.P. GEFFRAULT)

1. FONCTION GENERALE DU PROGRAMME

La structure des calculs de ce programme est liée au paramétrage adopté de l'arbre des classifications au moyen d'une représentation polonaise(*). Rappelons que cette dernière est une suite d'entiers positifs ou négatifs : un entier positif définit le code d'un élément de l'ensemble à classer et un entier négatif, un opérateur de réunion de deux classes ; la valeur absolue de cet entier négatif représentant le niveau de réunion.

Relativement à un tableau de données numériques : Individus ou Objets x Variables numériques, on suppose disposer de différentes classifications hiérarchiques (deux ou trois en général) sur l'ensemble des individus, sous la forme d'arbres "condensés" des classifications (cf.(*)), respectivement codés au moyen de leur représentation polonaise. La première classification hiérarchique est généralement établie pour une description des individus à travers l'ensemble V de toutes les variables et les suivantes, pour une description à travers, respectivement, des sous-ensembles de V. On suppose également établie une classification hiérarchique sur l'ensemble V des variables dont on retient un ou quelques niveaux parmi les plus significatifs. On suppose également avoir retenu pour chacun des arbres de classification sur l'ensemble des individus ou objets, deux ou trois niveaux parmi les plus significatifs.

Le but du programme est, relativement à une même classification définie par l'un des niveaux de l'un des arbres de classification ci-dessus mentionnés, d'"expliquer" la raison de formation des classes et de représenter chacune d'entre elles par un élément type.

A cette fin, on calcule pour chacune des classes existantes à un niveau donné, les moyennes, écarts-types, maxima et minima des différentes variables composant V, ainsi qu'un élément type par classe. On calcule d'autre part, pour chacune des variables, un "coefficient de discrimination" de la classification par la variable, lequel est basé sur une décomposition de la variance de la variable en variance intra-classes et en variance inter-classes. Toujours à partir de cette même formule de décomposition, on calcule le "coefficient de responsabilité" de chacune des variables dans la formation de chacune des classes de la partition étudiée. Le plus original est la détermination de ce "coefficient de responsabilité" d'une classe de mesures (i.e. de variables) dans la formation de chacune des classes d'objets ; cette classe de variables est dans la pratique l'une de celles obtenues à un niveau "significatif" de la classification hiérarchique de l'ensemble V, elle apparaît généralement comme caractérisant la description d'un aspect de l'objet.

Le calcul est décomposé par arbre de classification de l'ensemble des objets (codé au moyen de sa représentation polonaise) et pour la suite croissante des quelques niveaux de l'arbre retenus comme étant les plus significatifs. Cette démarche permet d'éviter de reprendre certains calculs lorsqu'on examine les classes d'un niveau, pour des classes déjà rencontrées à un niveau antérieur.

2. CALCULS EFFECTUES

Chaque élément x de l'ensemble E à classer peut être représenté par un point de $R_{card V}$, de coordonnées $\{v(x)/v \in V\}$ et affecté de la masse $\frac{1}{n}$, où V désigne l'ensemble des variables descriptives, $v(x)$ la mesure de la variable v sur l'individu x , n est le cardinal de E . Le moment d'inertie du nuage obtenu est :

$$I = \sum_{x \in E} \frac{1}{n} (\sum_{v \in V} (v(x) - \bar{v})^2)$$

si l'espace de représentation est muni de la distance euclidienne. La part de I due à une seule variable v est la variance de cette variable, qui peut être décomposée en somme de la variance intra-classes et de la variance inter-classes d'une partition $\{E_r/r = 1 \text{ à } k\}$ de E :

$$\frac{1}{n} \sum_{x \in E} (v(x) - \bar{v})^2 =$$

$$\sum_{1 \leq r \leq k} \frac{n_r}{n} \left(\frac{1}{n_r} \sum_{x \in E_r} (v(x) - \bar{v}(r))^2 \right) + \sum_{1 \leq r \leq k} \frac{n_r}{n} (\bar{v}(r) - \bar{v})^2$$

$$n_r = \text{card } E_r$$

$\bar{v}(r)$: moyenne de v dans la classe E_r

Le coefficient $C(v)$:

$$C(v) = \frac{\sum_{r=1 \text{ à } k} \frac{n_r}{n} (\bar{v}(r) - \bar{v})^2}{\frac{1}{n} \sum_{x \in E} (v(x) - \bar{v})^2} \quad (1)$$

est le rapport de la variance inter-classes à la variance globale. $C(v)$ sera d'autant plus proche de 1 que la variable v expliquera ou discriminera la partition considérée.

La part du coefficient $C(v)$, due à une classe E_r de la partition, est la part d'inertie relative à cette classe E_r et à la variable v :

$$C^r(v) = \frac{\frac{n_r}{n} (\bar{v}(r) - \bar{v})^2}{\frac{1}{n} \sum_{x \in E} (v(x) - \bar{v})^2}$$

$C^r(v)$ sera d'autant plus proche de 1 que la moyenne de v dans la classe E_r s'écartera de la moyenne globale. Il pourra, ainsi, exprimer la responsabilité de v dans la formation de la classe E_r , ou l'originalité de cette classe par rapport à la mesure v .

Soit une partition $\{V_s/s = 1 \text{ à } h\}$ de V , obtenue par exemple, grâce à une classification des variables. L'inertie du nuage due à une classe V_s s'exprime :

$$\sum_{x \in E} \frac{1}{n} \left(\sum_{v \in V_s} (v(x) - \bar{v})^2 \right)$$

En reprenant la décomposition de la variance, il est possible de définir la discrimination de la classe V_s pour une partition $\{E_r/r=1 \text{ à } k\}$ de E , ainsi :

$$C(V_s) = \frac{\sum_{v \in V_s} \sum_{1 \leq r \leq k} \frac{n_r}{n} (\bar{v}(r) - \bar{v})^2}{\sum_{v \in V_s} \sum_{x \in E} \frac{1}{n} (v(x) - \bar{v})^2} \quad (2)$$

On peut également envisager la contribution de la classe E_r et des variables de V_s , d'où on "coefficient de responsabilité de la classe V_s " dont l'expression est :

$$C^r(V_s) = \frac{\sum_{v \in V_s} \frac{n_r}{n} (\bar{v}(r) - \bar{v})^2}{\sum_{v \in V_s} \text{var}(v)} \quad (3)$$

Dans le programme, on calcule les coefficients $C(v), C^r(v)$, pour plusieurs partitions, et pour toutes les variables mesurées (pas seulement celles intervenant dans la classification hiérarchique considérée). D'autre part, les coefficients $C(V_s), C^r(V_s)$ et $C(V_s)$ pour une partition de l'ensemble V .

Dans chaque classe C l'élément-type est celui qui réalise :

$$\min_{x \in C} \sum_{y \in C} [d(x,y)]^2$$

$d(x,y)$ étant la distance euclidienne entre deux éléments x et y de la classe C , c'est-à-dire :

$$(d(x,y))^2 = \sum_{v \in V'} (v(x) - v(y))^2$$

où seules les variables intervenues dans la classification (sous-ensemble V' de V) sont prises en compte pour déterminer l'élément-type.

Pour nous fixer les idées, avant d'aborder la suite, rappelons les différentes fonctions du programme :

- pour chaque variable descriptive, calcul de la moyenne et de l'écart-type,
- reconstitution d'une partition de l'ensemble des variables obtenue par une classification hiérarchique de ces variables,
- pour chaque partition, calcul des "coefficients de discrimination" des différentes variables et des classes de variables.
- pour chaque classe :
 - * moyennes et écarts-types des différentes variables dans la classe,
 - * maxima et minima de ces variables,
 - * recherche d'un élément-type,
 - * calcul des "coefficients de responsabilité" des diverses variables et des classes de variables.

3. ORGANISATION DU FICHIER DES DONNES

La structure des trois fichiers utilisés intervient considérablement tout au long du programme :

- un premier fichier (file 10) contient le nombre d'individus (IO), le nombre de variables (JO), les noms des individus (tableau NOM) et les valeurs des mesures sur chaque individu (tableau VAL) :

IO	
JO	
NOM(1)	VAL(1,1) ... VAL(1,J) ... VAL(1,JO)
⋮	⋮ ⋮ ⋮
NOM(I)	VAL(I,1) ... VAL(I,J) ... VAL(I,JO)
⋮	⋮ ⋮ ⋮
NOM(IO)	VAL(IO,1)... VAL(IO,J)... VAL(IO,JO)

On remarquera une différence de notations dans ce programme par rapport à la plupart des autres programmes où on désignait par CARDI (au lieu de IO) et par CARDJ (au lieu de JO) les cardinaux respectifs de l'ensemble des individus et de celui des variables ; mais cela ne présente aucune gêne.

- un deuxième fichier de données (file 11) où l'on trouve les noms des variables (tableau NAM) ainsi que la "représentation polonaise" de la classification hiérarchique de ces variables (tableau MPOL de dimension LO) et le niveau de partition retenu pour cet arbre de classification (NPM, codé négativement) :

NAM(1)	
⋮	
NAM(J)	
⋮	
NAM(JO)	
MPOL(1)	MPOL(20)
.....	
.....	MPOL(LO)

- enfin, un troisième fichier (file 12) où on lit, pour chaque classification hiérarchique, un code identificateur de l'arbre de classification (NUC) un tableau (INDIC) indicatif des variables intervenues, le tableau de la représentation polonaise (IPOL de dimension KO), les niveaux que l'on désire étudier, codés par des entiers négatifs, et rangés par ordre décroissant (NP). Le traitement de chaque arbre de classification cesse quand NP est nul et le traitement général quand NUC est égal à 0.

NUC	INDIC(1) ... INDIC(J) ... INDIC(JO)
IPOL(1) IPOL(20)
 IPOL(KO)
NP	
NP	
:	
NP=0	
NUC	INDIC(1) ... INDIC(J) ... INDIC(JO)
.	.
.	.
NP=0	
NUC=0	

4. ORGANISATION DES CALCULS

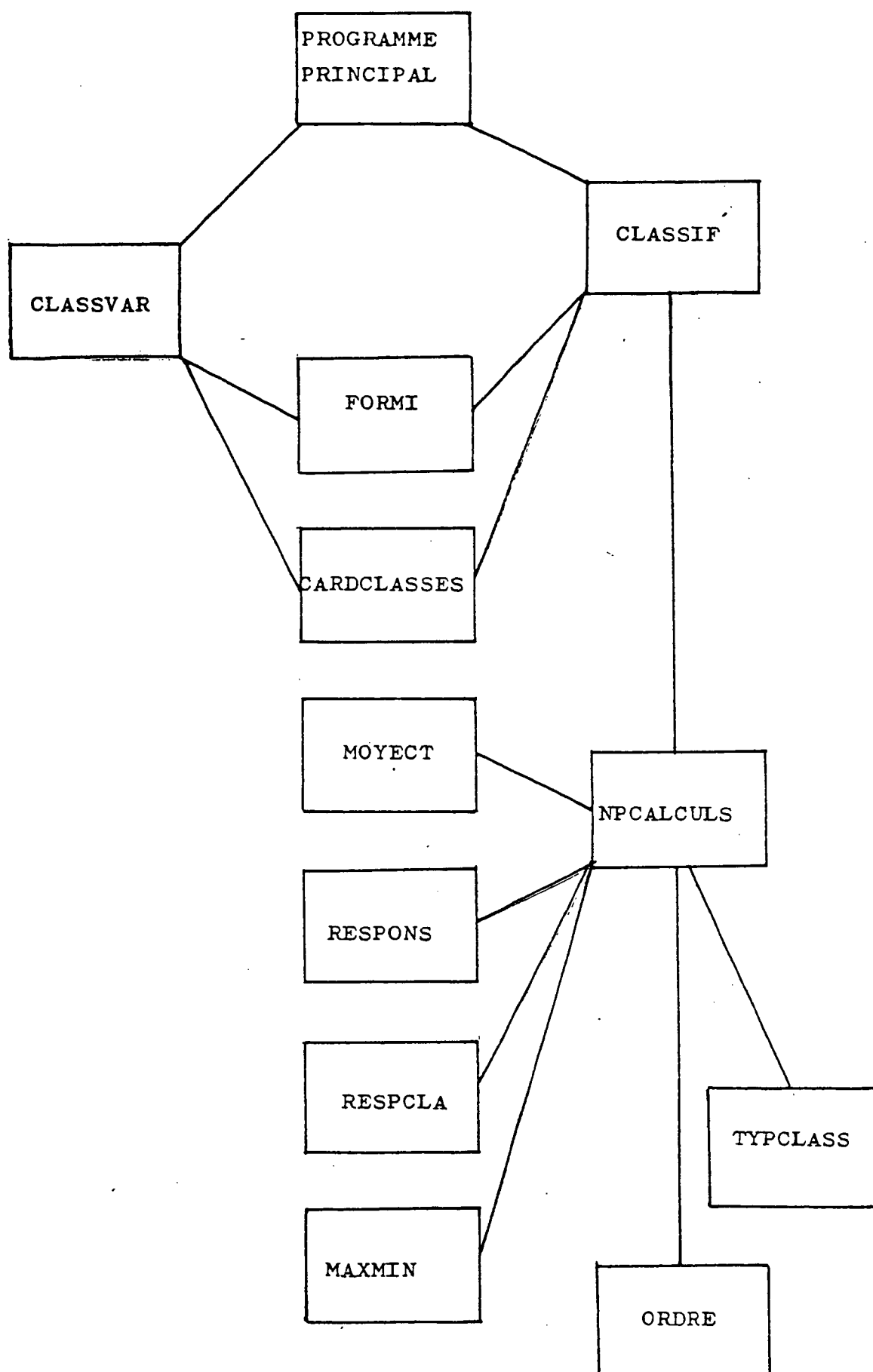


Figure 1 : Programmes d'interprétation des classifications hiérarchiques ; appel des sous-programmes.

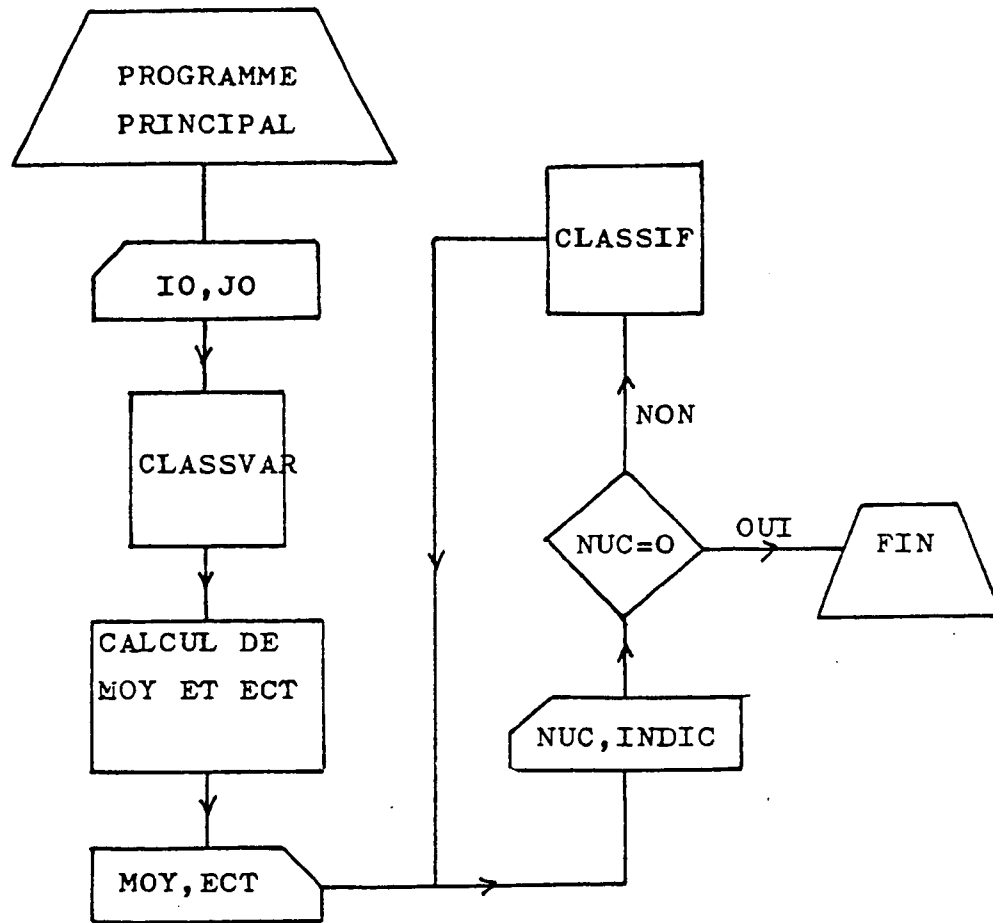


Figure 2 : Organigramme du programme principal

Programme principal et sous-programmes, dans leur déroulement

4.1. Programme principal

Il comprend :

- 1°) La lecture du nombre d'individus (IO) et du nombre de variables (JO) ;
- 2°) L'appel du sous-programme CLASSVAR (sans arguments) ;
- 3°) Le calcul des moyennes (MOY) et écarts-types (ECT) des différentes variables descriptives, puis l'écriture des résultats ;
- 4°) La lecture du code (NUC) d'un arbre de classification et du tableau INDIC associé, qui permettra de retrouver les variables utilisées pour établir la classification hiérarchique NUC ;
- 5°) L'arrêt du programme, si NUC est nul ; sinon, le passage à l'étape suivante ;
- 6°) L'appel du sous-programme CLASSIF (argument d'entrée : INDIC) ; puis le retour à l'étape 4.

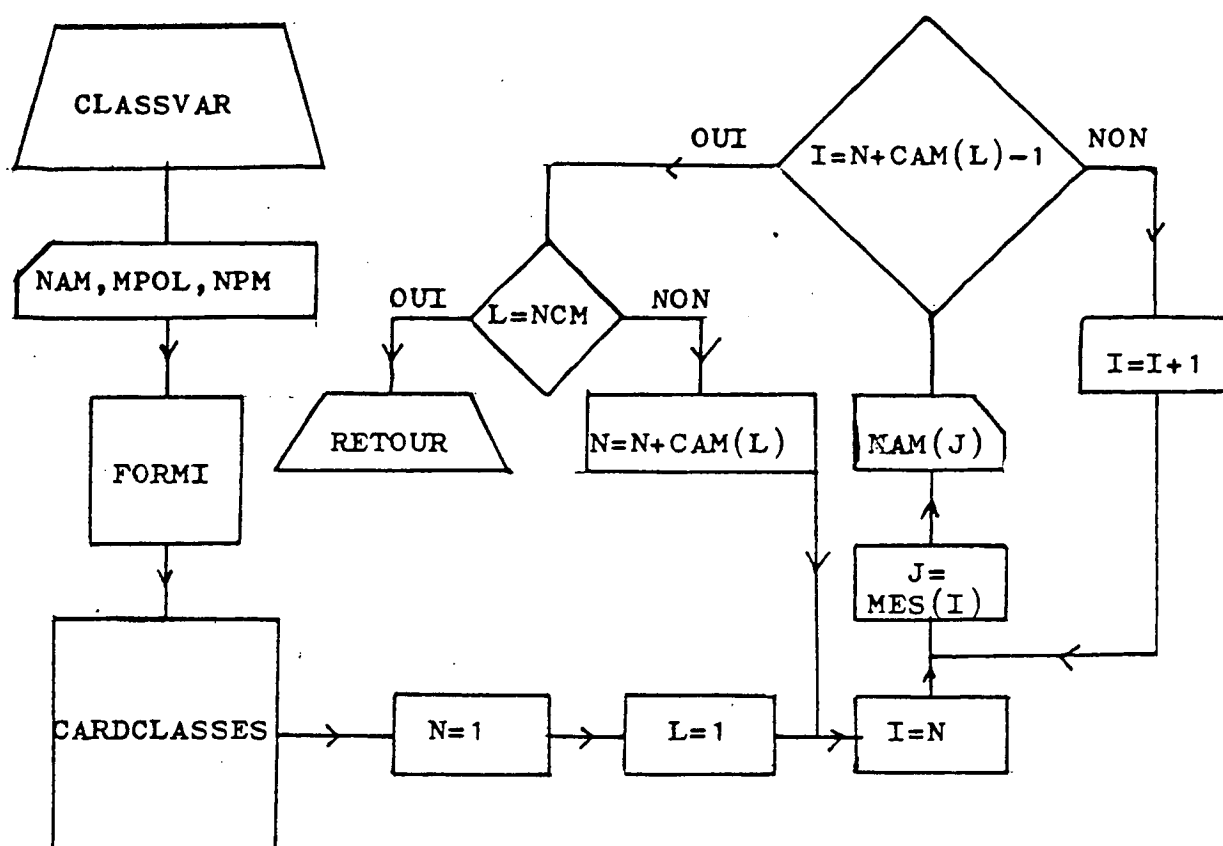


Figure 3 : Organigramme du sous-programme CLASSVAR

4.2. Sous-programme CLASSVAR

Après avoir lu les noms des variables (tableau NAM), le tableau MPOL, représentant la classification hiérarchique de ces variables (représentation "polonaise"), puis le niveau NPM de l'arbre correspondant à la partition choisie pour l'ensemble des variables, on fait appel aux sous-programmes FORMI et CARDCLASSES.

- . Arguments d'entrée du sous-programme FORMI :
 - MPOL
 - LO, nombre d'éléments dans le tableau MPOL ($LO=2*JO-1$)
 - JO
- . Argument de sortie du sous-programme FORMI :
 - MES tableau qui contient les codes (entiers positifs) des variables, rangés dans l'ordre où on les rencontre, si on lit le tableau de la représentation polonaise de gauche à droite.
- . Arguments d'entrée du sous-programme CARDCLASSES :
 - MPOL
 - LO
 - NPM
- . Arguments de sortie du sous-programme CARDCLASSES
 - CAM tableau qui contient les cardinaux des classes de variables dans l'ordre où on les rencontre en lisant MPOL de gauche à droite ;
 - NCM est le nombre de classes de variables ;
 - NFM contient les niveaux de formation des classes de variables.

Après l'exécution des sous-programmes FORMI et CARDCLASSES, il est possible d'imprimer, pour chaque classe de variables, les noms des variables qu'elle contient, en considérant que :

- MES(1) contient le code du premier élément de la première classe ;
- si MES(N) contient le premier élément de la L-ième, l'adresse du dernier élément de cette classe est $N + \text{CAM}(L) - 1$, et celle du premier élément de la classe suivante est $N + \text{CAM}(L)$ ($L < \text{NCM}$).

4.3. Sous-programme CLASSIF

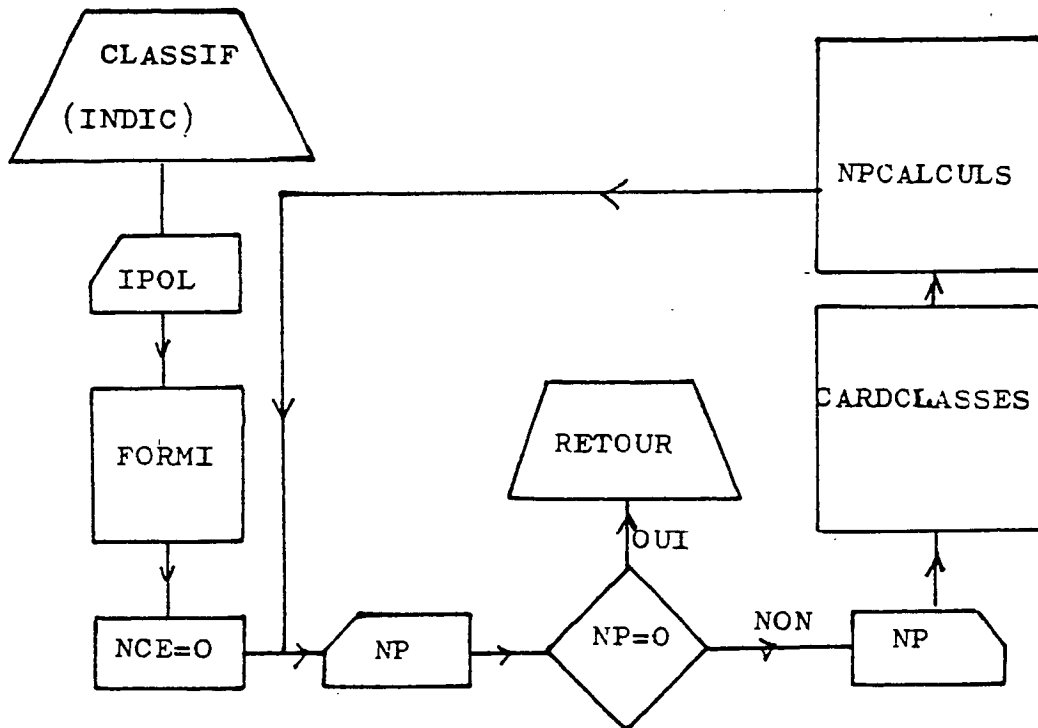


Figure 4 : Organigramme du sous-programme CLASSIF

Les différentes étapes du sous-programme CLASSIF sont :

- 1°) La lecture du tableau IPOL, représentation polonaise de la classification hiérarchique correspondant au tableau INDIC ;
- 2°) L'appel du sous-programme FORMI dont les arguments d'entrée sont :
 - IPOL
 - KO, nombre d'éléments dans le tableau IPOL ($KO = 2 * IO - 1$)
 - IO
 et I', argument de sortie :
 - INDIV, tableau qui contient les codes (entiers positifs) des individus classifiés, dans l'ordre avec lequel ils sont rangés dans le tableau de la représentation polonaise IPOL ;
- 3°) L'initialisation à 0 de NCE, qui à toutes les étapes du traitement de l'arbre de classification NUC, contient le nombre de classes différentes rencontrées dans des partitions de NUC précédemment étudiées : sa valeur changera quand une partition sera achevée ;
- 4°) La lecture de NP, un des niveaux choisis dans la hiérarchie de partitions NUC ;
- 5°) Si NP est nul, le retour au programme appelant (programme principal) ; sinon, passage à l'étape suivante ;

6°) L'appel du sous-programme CARDCLASSES, dont les arguments d'entrée sont :

- IPOL
- KO
- NP

et les arguments de sortie :

- CARD, tableau qui contient les cardinaux des classes, existant au niveau NP, dans l'ordre où on les rencontre en lisant IPOL de gauche à droite ;
- NC est le nombre de classes de la partition au niveau NP ;
- NF est un tableau qui contient les niveaux de formation des classes du niveau (codés négativement ou positivement pour des singletons).

7°) L'appel du sous-programme NPCALCULS, qui a pour arguments d'entrée :

- CARD
 - NC
 - NF
 - NCE
 - INDIC
 - NIV tableau qui conserve les niveaux de formation des classes qui ont été rencontrées à des niveaux antérieurs (à l'actuel NP) de la classification hiérarchique NUC ; au premier appel de NPCALCULS, NIV est vide ; les éléments de NIV sont des entiers positifs (singletons) ou négatifs (autres classes) ;
- Après l'exécution de NPCALCULS, NCE et NIV ont été modifiés :
- à NCE, on a ajouté le nombre de nouvelles classes rencontrées au niveau NP ;
 - le tableau NIV a été prolongé pour contenir les niveaux de formation de ces classes ;
- à la fin de cette étape, il y a retour à l'étape 4.
- N est le compteur qui permet la reconstitution de la classe (cf.§2) ;
 - M est le compteur des éléments de la classe L.

Il y a retour au programme appelant (CLASSIF ou CLASSVAR), quand le tableau IPOL est épuisé, c'est-à-dire quand K est supérieur à KO. (voir figure 1, organigramme).

4.4. Sous-programme NPCALCULS

Ce sous-programme commence par l'initialisation à 0 des coefficients de discrimination des mesures (DISC) et des classes de mesures (DEC). Ils seront calculés (voir paragraphe 2 ci-dessus) par incrémentation de chacune des valeurs de RESP (coefficient de responsabilité d'une variable) ou de REC (coefficient de responsabilité d'une classe de mesures) relatives aux différentes classes de la partition du niveau NP. Ensuite, on introduit quatre variables entières :

- INP (initialisé à 0) est, à toutes les étapes du sous-programme, le nombre de classes du niveau NP déjà rencontrées, mais qui n'ont pas été étudiées à des niveaux antérieurs à NP de la classification hiérarchique NUC ; quand la partition du niveau NP est achevée, INP contient le nombre de nouvelles classes découvertes au niveau NP ;
- L est le rang de la classe de la partition du niveau NP, que l'on est en train d'étudier (initialisé à 0) ;
- N est l'adresse, dans le tableau INDIV, du premier élément de la classe en cours d'étude (initialisé à 1) ;
- NK est l'adresse, dans le tableau INDIV, du dernier élément de la classe en cours d'étude.

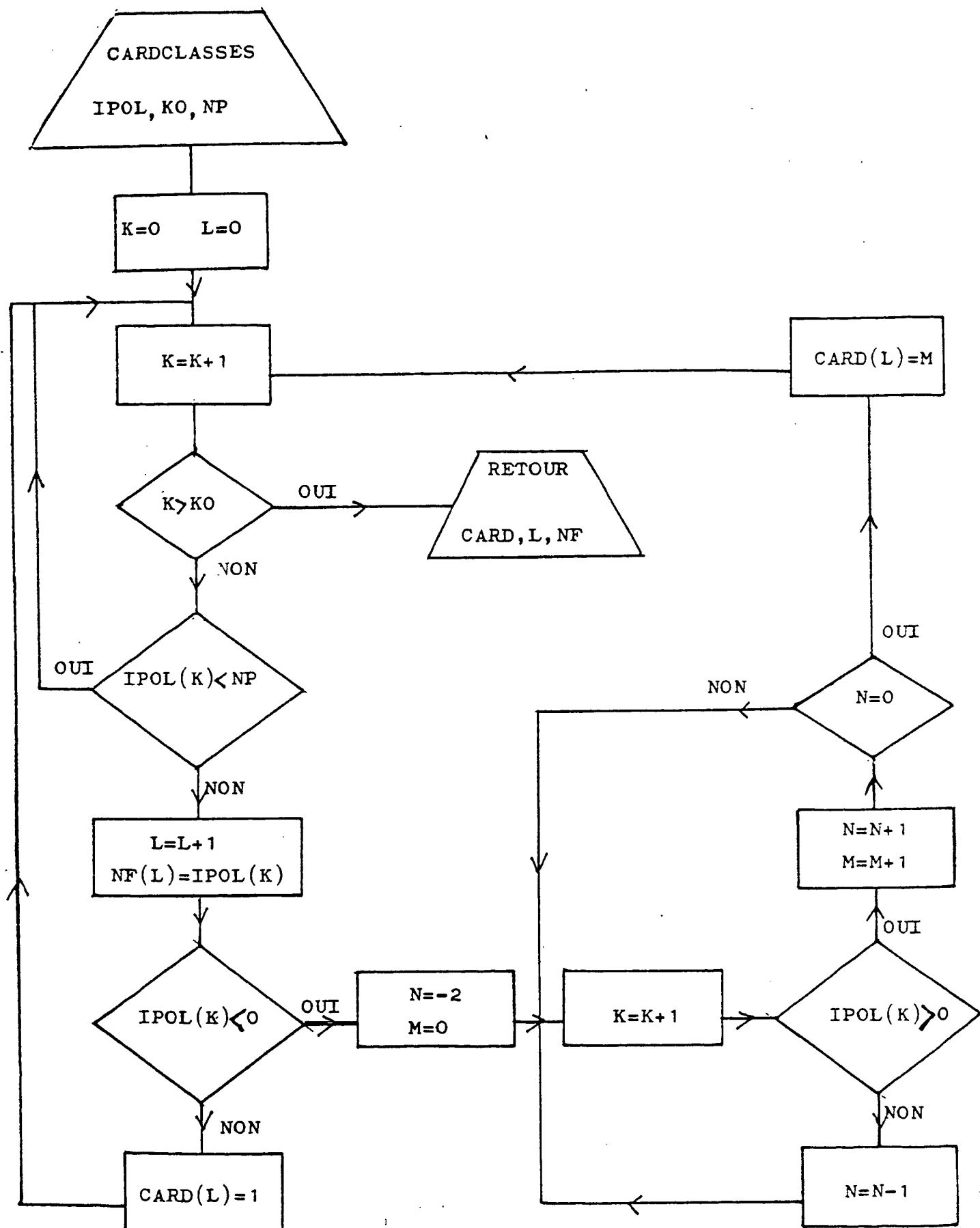


figure 5 : Organigramme du sous-programme CARDCLASSES

Pour chacune des classes de la partition obtenue au niveau NP, on appelle les sous-programmes :

- MOYECT, dont les arguments d'entrée sont :
 - * N
 - * CARD(L): nombre d'éléments dans la L-ième classe de la partition du niveau NP ;

et les arguments de sortie :

- * MOI : tableau des moyennes des variables dans la L-ième classe ;
- * ECI : tableau des écarts-types des variables dans la L-ième classe ;

- RESPONS, qui a pour paramètres d'entrée :
 - * MOI
 - * CARD(L)
 - * DISC

et, pour paramètres de sortie :

- * RESP
- * DISC, dont les valeurs ont été modifiées ;
- RESPCLA, avec, comme paramètres d'entrée :
 - * MOI
 - * CARD(L)
 - * DEC

et, comme paramètres de sortie :

- * DEC (valeurs actualisées).

Dans le balayage ascendant des quelques niveaux retenus de l'arbre des classifications, une même classe L peut se retrouver à différentes reprises ; la question se pose de savoir s'il y a lieu de refaire tous les calculs relatifs à cette classe. Il est clair qu'il est inutile de déterminer à nouveau les maxima et minima des différentes variables, ainsi que l'élément type de la classe. Toutefois, afin d'éviter une complication algorithmique, on a choisi de refaire les calculs des moyennes, écarts-types, coefficients de responsabilité des différentes variables prises une à une ou des classes de variables ; en effet, il y avait en outre lieu de toute façon de calculer les valeurs des coefficients de discrimination de la nouvelle classification par chacune des variables (resp. des classes de variables).

Il est essentiel que le balayage de l'arbre se fasse de manière ascendante. En effet, pour se rendre compte si L est une "nouvelle" classe, on peut comparer son niveau de formation à ceux des classes déjà rencontrées dans une partition strictement précédente à celle qu'on examine ; car si le niveau de formation de L est inférieur ou égal à celui de la partition strictement précédente, c'est que la classe L est nécessairement l'une des classes de cette dernière partition.

Pour plus de détail et au risque de nous répéter, nous allons reprendre certains éléments des rôles de différents sous-programmes.

5. ROLE DES SOUS PROGRAMMES

5.1. Sous-programme FORMI

Son rôle consiste tout simplement à retenir dans la représentation polonaise de l'arbre des classifications, dans l'ordre où ils se présentent, la suite des entiers positifs, représentant la suite des codes des éléments de l'ensemble à classer. Cet ordre des éléments est celui latéral des terminaux ou feuilles de l'arbre des classifications. C'est le tableau ITEM à une dimension (tableau ligne) qui recueille cette suite d'entiers. De façon précise, les paramètres formels IPOL, KO, ITEM et IO sont associés aux paramètres effectifs MPOL, LO, MES et JO dans le cas de la classification hiérarchique des variables et à IPOL, KO, INDIV et JO dans le cas de celle des individus.

5.2. Sous-programme CARDCLASSES

Pour retrouver à partir du tableau ITEM (MES dans le cas de la classification des variables et INDIV dans celui des individus) la partition à un niveau donné de l'arbre, il suffit de connaître la suite des cardinaux de la suite des classes de la partition considérée, telles qu'elles se présentent, de gauche à droite, pour la représentation polonaise de l'arbre des classifications et donc aussi, pour une représentation ascendante de ce dernier.

Pour reconnaître, à partir de la représentation polonaise, les classes formées pour la première fois à un niveau v de l'arbre ; il suffit de repérer dans la lecture de gauche à droite, le premier entier ($-v$), d'initialiser un compteur à -2 et de poursuivre la lecture jusqu'à ce que ce compteur soit nul, en ajoutant $+1$ (resp. -1) quand on rencontre un code positif (resp. négatif). Tous les nombres entiers positifs rencontrés dans cette lecture, représentent les éléments de la première classe recherchée. On poursuit la lecture de la représentation polonaise de l'arbre jusqu'à la rencontre du premier entier négatif ($-v$) et on recommence le processus pour délimiter la deuxième classe formée pour la première fois au niveau v ; et ainsi de suite...

Pour reconnaître maintenant la suite des classes de la partition déterminée par le niveau v de l'arbre des classifications, il suffit de procéder comme on vient de l'exprimer ; mais pour tout entier négatif de valeur absolue inférieure ou égale à v , qui correspond en effet à un noeud qui soustend une classe présente au niveau v .

En fait, on n'a besoin ici de retenir de la délimitation d'une classe que son cardinal ; c'est pour cette raison qu'on opère de la façon suivante :

(a) lecture du tableau IPOL de la représentation polonaise jusqu'à la découverte d'un élément supérieur ou égal à NP (entier négatif dont la valeur absolue est l'indice du niveau), ou jusqu'à épuisement.

(b) si le nombre rencontré est négatif, décompte des éléments de la classe formée à ce niveau ; sinon, la classe contient un seul élément. Après l'exécution de (b), on reprend la lecture du tableau IPOL (étape (a)) à partir du dernier élément de la classe dont on vient de calculer le cardinal ; de la sorte, on évite de rencontrer les niveaux de formation de classes, constituées à des niveaux antérieurs à INPI, mais englobées dans des classes déjà achevées au niveau étudié.

Pour être complet, précisons les notations du sous-programme
CARDCLASSES :

- NP : entier négatif dont la valeur absolue définit le niveau de la partition (paramètres effectifs NPM et NP),
 - K : adresse de l'élément courant du tableau IPOL,
 - L : rang de la classe en cours de constitution au niveau (-NP).
- Quand IPOL se trouve épuisé, L contient le nombre de classes de la partition (paramètres effectifs NC et NCM)
- M est le compteur des éléments de la classe L ; le tableau CARD permet la rétention des cardinaux des classes (paramètres effectifs CAM et CARD),
 - NF est le tableau où sont rangés, avec un codage négatif, les niveaux de formation des classes (paramètres effectifs NFM et NF).

5.3. Sous-programme CLASSVAR

Le rôle de ce sous-programme a déjà été complètement précisé dans la description séquentielle du programme. Il en est également de même pour le

5.4. Sous-programme CASSIF

5.5. Sous-programme NPCALCULS.

Il s'agit du sous-programme le plus important en ce sens qu'il réalise effectivement les calculs visés (cf. formules du paragraphe 2). Il a pour rôle de charger les tableaux suivants (initialement remis à zéro) :

DISC(JO), DEC(NCM), RESP(JOxNCE) et REC(NCMxNCE),

dont les tailles ont été indiquées entre parenthèses et qui, rappelons-le, contiendront en fin de programme, les coefficients de discrimination de la classification de l'ensemble des individus par les différentes variables, prises une à une (tableau DISC) et par les différentes classes de variables retenues (tableau DEC), les coefficients de responsabilité des différentes classes d'individus par les différentes variables (tableau RESP) et par les classes de variables retenues.

Pour chacune des classes du niveau -NP, on appelle les sous-programmes suivants :

5.5.1. MOYEYCT

Ce programme permet le calcul des moyennes (MOI) et des écarts-types (ECI) des différentes variables à l'intérieur des différentes classes. Dans le calcul IV désigne le numéro d'ordre du Ième élément du tableau INDIV et M, le cardinal de la classe ; si M est égal à 1, les moyennes sont les valeurs des mesures sur l'unique élément de la classe et les écarts-types sont nuls.

5.5.2. RESPONS

Calcul des coefficients de responsabilité et actualisation de la formule (1) du paragraphe (2) ; soit, calcul du chargement de DISC.

5.5.3. RESPCLA

Calcul des coefficients de responsabilité des classes de variables, tels qu'ils sont donnés par la formule (3) du paragraphe 2 ci-dessus : après multiplication par IO, on calcule numérateur et dénominateur (UNUM et VDEN) en parcourant la classe V_s ; ensuite, on modifie les valeurs du ta-

bleau DEC (cf. formule (2) § 2).

De la même manière que dans le sous-programme CLASSVAR, le compteur N est utilisé pour retrouver dans le tableau MES le début de chaque classe de variables.

5.5.4. MAXMIN

Les maxima (MAX) et minima (MIN) des différentes variables dans chaque classe d'objets de la partition étudiée, sont calculés dans ce programme. IV (resp. IW) désigne le numéro d'ordre du N-ième (resp. I-ème) du tableau INDIV.

5.5.5. ORDRE

Ce programme effectue le tri des éléments du tableau RESP par ordre décroissant des valeurs du coefficient. RI conserve les positions initiales de chaque valeur du coefficient. J variant de 1 à (JO-1), JM est l'indice du plus grand nombre rencontré parmi RESP(J+1),...,RESP(JO). TRANS et KRANS permettent les permutations dans RESP et RI.

5.5.6. TYPCLASS

Ce programme n'est exécuté que si la classe comporte plus de deux éléments.

On commence par calculer les critères à minimiser CRIT(somme des carrés des distances euclidiennes d'un élément à tous les autres éléments de la classe). CRIT(KY) (resp. CRIT(KZ)) désigne la valeur du critère pour le KYième (resp. KZième) de la classe L.KY et KZ correspondent aux Iième et Ilième du tableau INDIV, IV et IW étant les numéros d'ordre initiaux de ces éléments (ceux figurant dans la représentation polonaise). On peut remarquer que les expressions $DIF = (VAL(IV,J) - VAL(IW,J))^2$, J décrivant l'ensemble des mesures utilisées dans la classification NUC interviennent dans les calculs de CRIT(KY) et CRIT(KZ). On peut, ainsi, obtenir les valeurs définitives des critères en passant en revue toutes les paires d'éléments de L et en actualisant à chaque étape les valeurs de deux critères.

Ensuite, il reste à déterminer l'élément-type ; KPRO (numéro d'ordre dans la classe de l'individu minimisant CRIT et KP le numéro d'ordre initial de cet élément.

Quand toutes les classes ont fini d'être examinées, on ajoute à NCE le nombre de nouvelles classes trouvées au niveau NP(INP dans le programme) et on utilise à nouveau ORDRE pour ranger par valeurs décroissantes les coefficients de discrimination, avant impression. Le traitement de la partition de niveau NP s'achève tout à fait par l'édition des valeurs des coefficients de discrimination par les classes de mesures (tableau DEC).

6. LISTAGE DU PROGRAMME

```

      INTEGER NOM(102)
      INTEGER NAM(44)
      INTEGER INDIC(44)
      REAL VAL(102,44),MOY(44),ECT(44)
      COMMON /BLOC1/ KO
      COMMON /COM1/ JO
      COMMON /COM2/ VAL
      COMMON /COM3/ MOY,ECT
      COMMON /COM4/ IO
      COMMON /COM5/ NOM
      COMMON /COM10/ NAM
      READ (10,1004) IO,JO
1004 FORMAT (I4)
      KO=2*IO-1
      CALL CLASSVAR
      DO 1050 I=1,IO
1050 READ (10,1054) NOM(I),(VAL(I,J),J=1,JO)
1054 FORMAT (I4,2X,20F4.0/8X,16F4.0/8X,8F4.0/)
C CALCUL DES MOYENNES ET ECART.TYPES GLOBAUX
      DO 1070 J=1,JO
      MOY(J)=0
      ECT(J)=0
      DO 1060 I=1,IO
      MOY(J)=MOY(J)+VAL(I,J)
      ECT(J)=ECT(J)+(VAL(I,J))**2
1060 CONTINUE
      MOY(J)=MOY(J)/IO
      ECT(J)=((ECT(J)/IO)-(MOY(J)**2))**0.5
1070 CONTINUE
      WRITE (15,1078) (NAM(J),MOY(J),ECT(J),J=1,JO)
1078 FORMAT (9X,'MOYENNE',3X,'ECART-TYPE',5X/(2X,A3,2X,E10.4,2X,E10.
14))
1080 READ (12,1089) NUC,(INDIC(J),J=1,JO)
1089 FORMAT (A4,44I1)
      IF (NUC.EQ.0) STOP
      WRITE (15,1095) NUC
1095 FORMAT (//5X,'TRAITEMENT',X,'CLASSIFICATION',2X,A8//)
      CALL CLASSIF (INDIC)
      GO TO 1080
      END

```

CCC

```

      SUBROUTINE CLASSIF (INDIC)
      INTEGER NIV(100)
      INTEGER CARD(35),NF(35)
      DIMENSION IPOL(203)
      INTEGER INDIC(44)
      DIMENSION INDIV(102)
      COMMON /BLOC1/ KO
      COMMON /COM4/ IO
      COMMON /COM8/ INDIV
      READ(12,1102) (IPOL(K),K=1,KO)
1102 FORMAT (20I4)
      CALL FORMI (IPOL,KO,INDIV,IO)
      NCE=0
1110 READ (12,1117) NP
      IF (NP.EQ.0) RETURN
      WRITE (15,1115) NP
1115 FORMAT (* INTERPRETATION NIVEAU *,I4)
      CALL CARDCLASSES (IPOL,KO,CARD,NC,NP,NF)
      CALL NPCALCULS (CARD,NC,NF,NCE,NIV,INDIC)
      GO TO 1110
1117 FORMAT (I4)
      END

```

CCC

```

      SUBROUTINE CARDCLASSES (IPOL,KO,CARD,L,NP,NF)
      DIMENSION IPOL(KO)
      INTEGER CARD(35),NF(35)
      K=0
      L=0
10 K=K+1
      IF (K.GT.KO) RETURN
      IF (IPOL(K).LT.NP) GO TO 10
      L=L+1
      NF(L)=IPOL(K)
      IF (IPOL(K).LT.0) GO TO 30
      CARD(L)=1
      GO TO 10
30 N=-2
      M=0
40 K=K+1
      IF (IPOL(K).GT.0) GO TO 60
      N=N-1
      GO TO 40
60 N=N+1
      M=M+1
      IF (N.EQ.0) GO TO 80
      GO TO 40
80 CARD(L)=M
      GO TO 10
100 CONTINUE
      END

```

C

C

```

SUBROUTINE NPCALCULS (CARD,NC,NF,NCE,NIV,INDIC)
REAL DEC(15)
INTEGER NIV(100)
REAL RESP(44),DISC(44)
INTEGER RI(44)
INTEGER NAM(44)
COMMON /COM10/ NAM
COMMON /COM12/ NCM
REAL MOI(44),ECI(44)
REAL MAX(44),MIN(44)
INTEGER CARD(35),NF(35)
INTEGER INDIC(44)
COMMON /COM1/ JO
COMMON /COM6/ NK
DO 110 J=1,JO
110 DISC(J)=0
DO 115 L=1,NCM
115 DEC(L)=0
INP=0
L=0
N=1
120 L=L+1
WRITE(15,121) L,NF(L),CARD(L)
121 FORMAT (2X,"CLASSE NO ",I4," FORMEE AU NIVEAU ",I4," CONTENANT
1",I4,"ELEMENTS")
NK=N+CARD(L)-1
CALL MOYECT (N,CARD(L),MOI,ECI)
CALL RESPNS (MOI,CARD(L),RESP,DISC)
CALL RESPCLA (MOI,CARD(L),DEC)
I=1
130 IF (I.GT.NCE) GO TO 150
IF (NF(L).NE.NIV(I)) GO TO 140
WRITE (15,131)
131 FORMAT (10X,"CLASSE DEJA RENCONTREE",)
GO TO 170
140 I=I+1
GO TO 130
150 INP=INP+1
NIV(NCE+INP)=NF(L)
CALL MAXMIN (N,CARD(L),MAX,MIN)
WRITE(15,156)
156 FORMAT(16X,"RESPONSABILITE",6X,"MOYENNE",3X,"ECART-TYPE",X,"MAX
11UM",5X,"MINIMUM")
CALL ORDRE (RESP,RI)
DO 160 J=1,JO
JR=RI(J)
WRITE(15,161) NAM(JR),INDIC(JR),RESP(J),MOI(JR),ECI(JR),MAX(JR),MI
1N(JR)
160 CONTINUE
161 FORMAT(5X,A3,I3,7X,F6.3,9X,E10.4,2X,E10.4,4X,F4.0,7X,F4.0)
IF (CARD(L).LE.2) GO TO 170
CALL TYPCLASS (N,CARD(L),INDIC)
170 IF (L.NE.NC) GO TO 180
NCE=NCE+INP
CALL ORDRE (DISC,RI)
WRITE(15,175)
175 FORMAT(10X,"DISCRIMINATION")
DO 176 J=1,JO
JR=RI(J)
176 WRITE(15,177) NAM(JR),INDIC(JR),DISC(J)
177 FORMAT (2X,A3,I3,4X,F6.3)
WRITE (15,179) (L,DEC(L),L=1,NCM)
179 FORMAT (" DISCRIMINATION DE LA ",I4," EME CLASSE ",15X,F6.3)
RETURN
180 N=N+CARD(L)
GO TO 120
END

```

CC

```

SUBROUTINE MOYECT (N,M,MOY,ECT)
REAL VAL(102,44)
REAL MOT(44),ECT(44)
DIMENSION INDIV(102)
COMMON /COM1/ JO
COMMON /COM2/ VAL
COMMON /COM6/ NK
COMMON /COM8/ INDIV
DO 300 J=1,JO
IF (M.EQ.1) GO TO 295
MOT(J)=0
ECT(J)=0
DO 280 I=1,N,NK
IV=INDIV(I)
MOT(J)=MOT(J)+VAL(IV,J)
ECT(J)=ECT(J)+(VAL(IV,J)**2)
280 CONTINUE
MOT(J)=MOT(J)/M
ECT(J)=ECT(J)/M-MOT(J)**2
ECT(J)=ECT(J)**0.5
GO TO 300
295 IV=INDIV(N)
MOT(J)=VAL(IV,J)
ECT(J)=0
300 CONTINUE
RETURN
END

```

C

```

C
C
SUBROUTINE RESPON (MOI,M,RESP,DISC)
REAL RESP(44),DISC(44)
REAL MOY(44),ECT(44)
REAL MOI(44)
COMMON /COM1/ JO
COMMON /COM3/ MOY,ECT
COMMON /COM4/ IO
DO 400 J=1,JO
RESP(J)=M*(MOI(J)-MOY(J))*2/(IO*ECT(J))*2
DISC(J)=DISC(J)+RESP(J)
400 CONTINUE
RETURN
END

```

```

C
C
SUBROUTINE MAXMIN (N,M,MAX,MIN)
REAL MAX(44),MIN(44)
INTEGER INDIV(102)
REAL VAL(102,44)
COMMON /COM1/ JO
COMMON /COM2/ VAL
COMMON /COM6/ NK
COMMON /COM8/ INDIV
IV=INDIV(N)
DO 500 J=1,JO
MAX(J)=VAL(IV,J)
MIN(J)=VAL(IV,J)
IF (M.EQ.1) GO TO 500
DO 470 I=N+1,NK
IW=INDIV(I)
IF (VAL(IW,J).GE.MAX(J)) GO TO 450
IF (VAL(IW,J).GE.MIN(J)) GO TO 470
MIN(J)=VAL(IW,J)
GO TO 470
450 MAX(J)=VAL(IW,J)
470 CONTINUE
500 CONTINUE
RETURN
END

```

```

C
C
SUBROUTINE TYPCLASS (N,M,INDIC)
REAL CRIT(100)
INTEGER INDIC(44)
INTEGER INDIV(102),NOM(102)
REAL VAL(102,44)
COMMON /COM1/ JO
COMMON /COM2/ VAL
COMMON /COM5/ NOM
COMMON /COM6/ NK
COMMON /COM8/ INDIV
DO 510 I=1,M
CRIT(I)=0
DO 550 I=N+1,NK
IV=INDIV(I)
KY=i-N+1
DO 550 IL=N,I-1
IW=INDIV(IL)
KZ=IL-N+1
DO 550 J=1,JO
IF (INDIC(J).EQ.0) GO TO 550
DIF=(VAL(IW,J)-VAL(IV,J))*2
CRIT(KY)=CRIT(KY)+DIF
CRIT(KZ)=CRIT(KZ)+DIF
550 CONTINUE
KPRO=1
DO 570 I=2,M
IF (CRIT(KPRO).LT.CRIT(I)) GO TO 570
KPRO=I
570 CONTINUE
KP=INDIV(KPRO+N-1)
WRITE (15,581) NOM(KP)
581 FORMAT (2X,'ELEMENT-TYPE',X,'=',X,14///)
RETURN
END

```

```

C
C
SUBROUTINE ORDRE (RESP,RI)
INTEGER RI(44)
REAL RESP(44)
COMMON /COM1/ JO
DO 630 J=1,JO
RI(J)=J
DO 700 J=1,JO-1
JM=J
DO 670 K=J+1,JO
IF (RESP(K).LE.RESP(JM)) GO TO 670
JM=K
670 CONTINUE
TRANS=RESP(J)
RESP(J)=RESP(JM)
RESP(JM)=TRANS
KRANS=RI(J)
RI(J)=RI(JM)
RI(JM)=KRANS
700 CONTINUE
RETURN
END

```

C
C
C

```

SUBROUTINE FORMI (IPOL,K0,ITEM,IO)
DIMENSION IPOL(K0),ITEM(IO)
I=0
DO 800 K=1,K0
IF (IPOL(K).LT.0) GO TO 800
I=I+1
ITEM(I)=IPOL(K)
800 CONTINUE
RETURN
END

```

C
C
C

```

SUBROUTINE RESPCLA (MOI,M,DEC)
COMMON /COM3/ MOT,ECT
COMMON /COM4/ IO
REAL MOT(44),ECT(44)
REAL MOI(44),DEC(15)
COMMON /COM11/ MES,CAM
COMMON /COM12/ NCM
INTEGER CAM(35)
DIMENSION MES(44)
REAL RESP(44),REC(15)
N=1
DO 880 L=1,NCM
VNUM=0
VDEN=0
DO 870 I=N,N+CAM(L)-1
J=MES(I)
VNUM=VNUM+(MOI(J)-MOT(J))*2
VDEN=VDEN+ECT(J)*2
870 CONTINUE
REC(L)=M*VNUM/(IO*VDEN)
WRITE (15,875) L,REC(L)
875 FORMAT (" RESPONSABILITE DE LA ",I4," EME CLASSE ",10X,F6.3)
DEC(L)=DEC(L)+REC(L)
N=N+CAM(L)
880 CONTINUE
RETURN
END

```

C
C
C

```

SUBROUTINE CLASSVAR
INTEGER NAM(44)
DIMENSION MPOL(87),MES(44)
INTEGER CAM(35),NFM(35)
COMMON /COM1/ JO
COMMON /COM10/ NAM
COMMON /COM11/ MES,CAM
COMMON /COM12/ NCM
READ(11,1013) (NAM(J),J=1,JO)
1013 FORMAT (A3)
LO=2*JO-1
READ (11,1017) (MPOL(L),L=1,LO)
1017 FORMAT (20I4)
READ (11,1019) NPM
1019 FORMAT (I4)
CALL FORMI (MPOL,LO,MES,JO)
CALL CARDCLASSES (MPOL,LO,CAM,NCM,NPM,NFM)
N=1
WRITE (6,1022) NCM,NPM,LO
1022 FORMAT (" NCM=",I4," NPM=",I4," LO ",I4)
DO 1040 L=1,NCM
WRITE (15,1025) L
1025 FORMAT (5X,I4," EME CLASSE DE MESURES CONTENANT:")
DO 1030 I=N,N+CAM(L)-1
J=MES(I)
WRITE (15,1027) NAM(J)
1027 FORMAT (4X,A4)
1030 CONTINUE
N=N+CAM(L)
1040 CONTINUE
RETURN
END

```

II.2. DEGRE DE RESPONSABILITE DANS LE CAS D'UN TABLEAU D'INCIDENCE DES DONNEES

PROGRAMME RESPINC (T. CHANTREL)

1. FONCTION GENERALE DU PROGRAMME ET CALCULS EFFECTUES

La donnée est un tableau d'incidence indexé par Individus x Attributs, où on désignera par A l'ensemble des attributs de description et par I l'ensemble des individus.

On considère une partition $\{A_r / 1 \leq r \leq k\}$ de A, obtenue au moyen d'une classification par proximité de A et qui, pour nous dans la pratique, se trouve définie par un niveau "significatif" de l'arbre hiérarchique des classifications sur A.

Chaque classe A_r définit un profil d'attitude et la classification $\{A_r / 1 \leq r \leq k\}$, une typologie.

Associons à A le nuage de points de $\{0,1\}^n$

$$(A) = \{(\phi_a, \frac{1}{c(A)}) / a \in A\} \quad , \quad (1)$$

où $c(A)$ est le cardinal de A et où ϕ_a est le vecteur logique, caractéristique de la partie E_a :

$$(\phi_a(1), \phi_a(2), \dots, \phi_a(i), \dots, \phi_a(n)) \quad , \quad (2)$$

avec $\phi_a(i) = 1$ (resp. 0) si l'individu codé i possède l'attribut a (resp. sinon^a).

Munissons $\{0,1\}^n$ de la distance euclidienne ordinaire qui correspond d'ailleurs au cardinal de la différence symétrique quand on l'exprime au niveau de l'ensemble des parties de l'ensemble E des individus. En plongeant $\{0,1\}$ dans l'intervalle $[0,1]$; on a la formule suivante de décomposition de l'inertie totale du nuage, selon la partition de A :

$$\begin{aligned} \frac{1}{c(A)} \sum_{a \in A} ||\phi_a - \phi_A||^2 &= \frac{1}{c(A)} \sum_{1 \leq r \leq k} c(A_r) \left[\frac{1}{c(A_r)} \sum_{a \in A_r} ||\phi_a - \phi_{A_r}||^2 \right] \\ &+ \frac{1}{c(A)} \sum_{1 \leq r \leq k} c(A_r) ||\phi_{A_r} - \phi_A||^2, \quad (3) \end{aligned}$$

où $\phi_A(i)$ (resp. $\phi_{A_r}(i)$) est la i-ème composante de ϕ_A (resp. ϕ_{A_r})

L'inertie expliquée ou retenue par la classification est définie par le rapport

$$\sum_{1 \leq r \leq k} c(A_r) ||\phi_{A_r} - \phi_A||^2 \bigg/ \sum_{a \in A} ||\phi_a - \phi_A||^2 \quad , \quad (4)$$

où la contribution de la i-ème composante à ce rapport de variances est donnée par

$$\rho(i) = \sum_{1 \leq r \leq k} c(A_r) [\phi_{A_r}(i) - \phi_A(i)]^2 \bigg/ \phi_A(i) [1 - \phi_A(i)] \quad , \quad (5)$$

Cet indice "mesurera" le degré de responsabilité de l'individu i dans la formation des profils de comportement respectivement définis par les différentes classes A_r , $1 \leq r \leq k$.

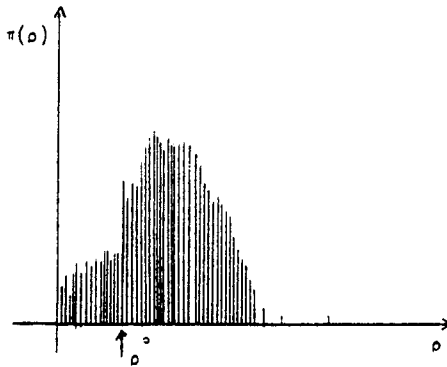
Le degré de responsabilité dans la formation de la seule classe A_r sera défini par

$$\rho_r(i) = c(A_r) \left[\phi_{A_r}(i) - \phi_A(i) \right]^2 / \phi_A(i) \left[1 - \phi_A(i) \right] \quad , (6)$$

L'analyse graphique de la distribution

$$\{\rho(i)/i \in I\} \quad , (7)$$

au moyen de l'histogramme de fréquences suivant où l'axe horizontal est celui des valeurs de $\rho(i)$ et où l'axe vertical représente la fréquence des sujets, montre en général une valeur ρ^0 correspondante à une crête de la distribution qui intervient après une plage de petites valeurs de ρ , faiblement représentés.



Dans ces conditions, un individu typique sera caractérisé par la relation

$$\rho(i) \geq \rho^0 \quad , (8)$$

$I^t = \{i/\rho(i) \geq \rho^0\}$ désignera l'ensemble des individus typiques.

La même analyse graphique de la distribution

$$\{\rho_r(i)/i \in I\} \quad , (9)$$

montre le même phénomène. ρ_r^0 étant l'analogue de ρ^0 ; soit

$$I_r = \{i/\rho_r(i) \geq \rho_r^0\} \quad , (10)$$

l'ensemble des individus "concernés" pour le profil de comportement défini par la classe A_r d'attributs.

Nous délimiterons un individu typique de la classe A_r au moyen de la relation

$$i \in I^t \cap I_r \quad , (11)$$

Soit maintenant H une partie quelconque de I, nous "mesurerons" le rôle de H dans la formation du profil de comportement A_r , par la valeur du rapport

$$\text{card}(H \cap I^t \cap I_r) / \text{card}(H), \quad (12)$$

qui est la proportion d'individus de la classe H qui sont typiques de la classe A_r .

La présentation ci-dessous est faite au moyen de deux programmes qui sont appelés à être regroupés en un seul.

Le premier programme calcule pour chaque individu (codé I dans le programme ($1 \leq I \leq \text{CARDI}$)) le coefficient de responsabilité globale $S(I)$, qui correspond à $\rho(i)$ ci-dessus (cf. formule (5)), ainsi que les coefficients de responsabilité partiels $\text{RESP}(I, K)$, $1 \leq K \leq \text{NBCL}$, qui correspondent aux $\rho_r(i)$ ci-dessus (cf. formule (6)) ; K désignant l'indice courant d'une classe d'attributs et NBCL, le nombre de classes d'attributs.

Ce programme édite le tableau ordonné des valeurs de la suite des lignes

$$(S(I), \text{RESP}(I, K) / 1 \leq K \leq \text{NBCL}), \quad (13)$$

où l'ordre est établi par valeurs décroissantes de $S(I)$. Le tri utilisé pour établir cet ordre est celui utilisé dans le programme de classification hiérarchique (cf. (*)) pour établir l'ordonnance.

Le deuxième programme calcule les degrés de responsabilité, mais dans une autre optique ; celle d'effectuer le tri d'une classe d'individus et son énumération. On suppose déjà établis les différents seuils : global ρ^0 (cf. (8) ci-dessus), noté SEUIL dans le programme, et locaux ρ_r^0 (cf. (10) ci-dessus), notés PSEUIL(K), $1 \leq K \leq \text{NBCL}$, dans le programme.

Avec les notations ci-dessus (cf. formules (10), (11) et (12)), il s'agit de $\text{card}(H \cap I^t \cap I_r)$ où le sous ensemble H d'individus se trouve défini par la possession d'une même modalité d'une variable qualitative exogène, aux modalités exclusives. NBITEM désigne dans le programme le nombre de modalités d'une telle variable et si H_J désigne la classe des sujets possédant la J-ème modalité ; le programme établit en fait la suite des cardinaux $\{\text{card}(H_J \cap I^t \cap I_r) / 1 \leq J \leq \text{NBITEM}\}$. Le tableau des cardinaux obtenu est désigné dans le programme par

$$\{ \text{TN}(J, K) / 1 \leq J \leq \text{NBITEM}, 1 \leq K \leq \text{NBCL} \}, \quad (14)$$

auquel on peut très aisément substituer un tableau de proportions conformément à la formule (12) ci-dessus.

2. PARAMETRES ET ORGANISATION DES CALCULS

CARDI (resp. CARDJ) désigne le cardinal de l'ensemble des individus (resp. variables). On lit, comme il est d'usage, la description du fichier individu I par individu I et c'est le tableau à une ligne VAR qui contient la description codée en zéros et uns d'un même individu à travers la suite

(*) "Programme de classification hiérarchique", I.C. Lerman, Publication IRISA n° 148, Juin 1981.

des attributs-modalités de différentes variables qualitatives ou, tout simplement, à travers une suite d'attributs de description. On désigne par NBCL le nombre de classes d'attributs retenues ; dont chacune définit un profil d'attitude. Il s'agit en fait d'une classification qui ne recouvre pas nécessairement l'ensemble de tous les attributs de description ; on peut en effet juger nécessaire, compte tenu notamment de la formation préalable de l'arbre des classifications sur l'ensemble des traits descriptifs, d'ignorer certains attributs.

Le calcul se trouve géré par un tableau ligne $A(J)$, $1 \leq J \leq \text{CARDJ}$, donnant pour chaque attribut codé J , le numéro K de la classe (d'attributs) dont il fait partie ; ce numéro est posé égal à 0 si l'attribut ne se trouve associé à aucune des classes retenues.

Signalons que le tableau B définit la suite des cardinaux des classes d'attributs retenues.

D'autre part, précisons que, relativement au deuxième programme, les modalités de la variable exogène se trouvent en tête de lecture.

Terminons en indiquant que $\text{SOM}(K)$ (resp. $\text{PROP}(K)$), $1 \leq K \leq \text{NBCL}$, désigne dans le programme le nombre (resp. la proportion) d'attributs de la classe K possédés par l'individu I en cours de traitement.

3. LISTAGE

```

DIMENSION A(164),VAR(164A),B(8)
DIMENSION SOM(8),PROP(8)
DIMENSION RESP(8622,8)
DIMENSION S(8622),IPERM(8622),IP(8622),IDEB(95),IFIN(95)
DIMENSION IDEB1(95),IFIN1(95)

C
READ(105,5) CARDI,CARDJ,NBCL
WRITE(108,1) CARDI,CARDJ,NBCL

C
C
LECTURE DES ADRESSES DES VARIABLES

C
READ(105,2)(A(J),J=1,CARDJ)
WRITE(108,2)(A(J),J=1,CARDJ)

C
C
LECTURE DES CARDINAUX DES CLASSES

C
READ(105,3)(B(J),J=1,NBCL)
NBLIGN=NBCL/8
NBR=NBCL-8*NBLIGN
WRITE(108,14)(J,B(J),J=1,8*NBLIGN)
IF(NBR.EQ.0) GO TO 99
WRITE(108,140) NBR,(J,B(J),J=8*NBLIGN+1,NBCL)
99 CONTINUE

C
C
LECTURE DES VARIABLES-INDIVIDUS PAR INDIVIDUS

C
I=1
12 READ(10,40)(VAR(J),J=1,CARDJ)
S(I)=0
TOT=0
DO 21 K=1,NBCL
SOM(K)=0
PROP(K)=0
21 CONTINUE
J=1
7 IF(A(J).EQ.0) GO TO 17
DO 41 K=1,NBCL
IF(A(J).EQ.K) SOM(K)=SOM(K)+VAR(J)
41 CONTINUE
17 J=J+1
IF(J.E.CARDJ) GO TO 7
DO 51 K=1,NBCL
PROP(K)=(1.*SOM(K))/B(K)
TOT=TOT+(PROP(K)*B(K)/CARDJ)
51 CONTINUE
DO 5 K=1,NBCL
RESP(I,K)=((B(K)/CARDJ)*((PROP(K)-TOT)*2))/((TOT*(1.-TOT))
5 CONTINUE

C
DO 61 K=1,NBCL
S(I)=S(I)+RESP(I,K)
61 CONTINUE

C
I=I+1
IF(I.LE.CARDI) GO TO 12

C
C
KL=CARDI
X=KL

C
DO 102 L=1,KL
IPERM(I)=L
102 CONTINUE
K2=SQRT(X)
LONG=KL/K2
KR=KL-K2*LONG
IDER(1)=1
IFIN(1)=LONG
IF(KR.EQ.0) PNT=LONG,M=1,N=K2,GO TO 10
IFIN(1)=KR
K2=K2+1,PNT=KT,M=1,N=1
10 CONTINUE
DO 55 K=2,K2
IDEB(K)=IFIN(K-1)+1
55 IFIN(K)=IFIN(K-1)+LONG
IDEB1(1)=1
IFIN1(1)=0
L=0
34 K3=SQRT(PNT)
LONG1=PNT/K3
KR1=PNT-K3*LONG1
IF(KR1.EQ.0) IFIN1(1)=IFIN1(1)+LONG1,GO TO 20
IFIN1(1)=IFIN1(1)+KR1
K3=K3+1
20 CONTINUE
DO 100 J=M,N
I=IDER(J)
IF(K3.EQ.1) L=L+LONG1,GO TO 31
DO 30 K=2,K3
IDEB1(K)=IFIN1(K-1)+1
30 IFIN1(K)=IFIN1(K-1)+LONG1
31 CONTINUE

```



```

DO 42 K=1,K3
42 CALL TRI(K,S,Iperm,IDEb1,IFIN1)
DO 53 TRI(K,S,Iperm,IDEb1,IFIN1)
53 IP(K)=Iperm(K)
LOGIC=1
IF(K3.EQ.1) GO TO 33
DO 50 K=2,K3
CALL INCLASS(K,LOGIC,S,Iperm,IP,IDEb1,IFIN1,L,I)
I=IDEb1(J)
IF(KR1.EQ.0) L=L-K*LONG1,GO TO 50
L=L-(K-1)*LONG1-KR1
50 CONTINUE
IF(LOGIC.EQ.1) GO TO 64
DO 52 K=I+1,IFIN1(K3)
52 Iperm(K)=IP(K)
64 CONTINUE
DO 66 K=I+1,IFIN1(K3)
66 IP(K)=Iperm(K)
L=IFIN1(K3)
33 CONTINUE
IF(IFIN1(K3).EQ.KL) GO TO 100
IDEb1(1)=IFIN1(K3)+1
IF(K3.EQ.1) GO TO 37
IFIN1(1)=IFIN1(K3)+LONG1
37 IF(KR1.EQ.0) GO TO 36
IFIN1(1)=IFIN1(K3)+KR1
36 CONTINUE
100 CONTINUE
IF(N.EQ.K2) GO TO 32
PNT=LONG,M=2,N=K2
IFIN1(1)=IFIN1(K3)
GO TO 34
32 CONTINUE
LOGIC=1
DO 65 K=2,K2
L=0
I=1
65 CALL INCLASS(K,LOGIC,S,Iperm,IP,IDEb,IFIN,L,T)
IF(LOGIC.EQ.1) GO TO 80
DO 81 K=1,KL
81 Iperm(K)=IP(K)
80 DO 82 K=1,KL
82 IP(K)=Iperm(K)
C
C C
C ECRITURE DU DEGRE DE RESPONSABILITE (20)
WRITE(108,101)
WRITE(108,200) NBCL,(K,K=1,NBCL)
DO 444 K=1,KL
WRITE(108,300) K,S(IP(K)),NBCL,(RESP(IP(K),KK),KK=1,NBCL)
444 CONTINUE
C
C
1 FORMAT(3I4)
2 FORMAT(40I2)
3 FORMAT(40I2)
40 FORMAT(50X,48I1,2X,28I1,1X,30I1,4X,6I1,416X,10I1,20X,9I1,18X,3
14 FORMAT(///30X,'CARDINAUX DES CLASSES DE LA CLASSIFICATION'/
130X,42(' ')/1X,8('CARDCL N',I2,'=',I2,2X))
140 FORMAT(1X,N('CARDCL N',I2,'=',I2,2X))
101 FORMAT(///30X,'MESURE DU DEGRE DE RESPONSABILITE'/30X,33(' '))
200 FORMAT(///20X,'DEGRE',N(4X,'CLAS N',I2))
300 FORMAT(///INDIVIDU N ',I4,4X,F8.3,2X,N(2X,F8.3,3X))
STOP
END

```


II.3. DEGRE DE RESPONSABILITE DANS LE CAS DE DONNEES DE CONTINGENCE

PROGRAMME RESPCONT (A. PROD'HOMME)

1. FONCTION GENERALE DU PROGRAMME

On suppose ici que la donnée est définie par une juxtaposition horizontale de tables de contingence, indexée par un ensemble de la forme $I \times (J^{(1)} \cup J^{(2)} \cup \dots \cup J^{(h)} \cup \dots \cup J^{(k)})$ où $I \times J^{(h)}$ indexe un vrai tableau de contingence, $1 \leq h \leq k$.

Une telle structure de l'information statistique est particulièrement fréquente dans les données à caractère géographique. Ainsi, le cas réel dans le cadre duquel ce programme a été mis au point est celui où I est formé d'un ensemble d'unités géographiques (communes, départements, etc...) et où $J^{(h)}$ est l'ensemble sans structure des modalités d'un caractère descriptif (on dit encore d'une variable qualitative nominale), $1 \leq h \leq k$.

On suppose établie une classification par proximité statistique (avec ou sans contrainte de connexité pour les classes formées) dans sa description à travers $H = J^{(1)} \cup J^{(2)} \cup \dots \cup J^{(k)}$. Notons $\{I_\ell / 1 \leq \ell \leq L\}$ une telle partition qui, dans le cadre de nos méthodes peut correspondre à un niveau "significatif" de l'arbre des classifications sur I (cf. chap.2 et 5 de "Classification et analyse ordinaire des données").

Le but de ce programme est d'établir pour chaque j_0 de H la valeur d'un coefficient de "responsabilité" (nous disons encore de "discrimination" ou bien d'"explication" de chacune des classes I_ℓ , $1 \leq \ell \leq L$, par j_0).

Nous allons rapidement rappeler le principe des calculs effectués ; pour plus de détails, on consultera le chapitre 3 de l'ouvrage cité ci-dessus.

Relativement à un même tableau de contingence $I \times J$:

$$\{k_{ij} / (i,j) \in I \times J\}, \quad (1)$$

on considère la représentation classique de I à travers J au moyen du nuage $N(I)$:

$$N(I) = \{(f_j^i, p_i) / i \in I\} \quad (2)$$

où f_j^i désigne le point $(f_1^i, f_2^i, \dots, f_m^i)$ de $R^m = (m = \text{card}(J))$, dont la suite des composantes est la suite des proportions conditionnelles

$$f_j^i = \frac{f_{ij}}{p_i} = \frac{k_{ij}}{k_{i.}},$$

$$\text{où } k_{i.} = \sum \{k_{ij} / j \in J\},$$

d'autre part, le poids p_i est défini par le rapport $k_{i.} / k_{..}$ où

$$k_{..} = \sum \{k_{ij} / (i,j) \in I \times J\},$$

enfin, l'espace ambiant R^m est supposé muni de la métrique du χ^2 :

$$q(e_j, e_h) = \begin{cases} 0 & \text{si } h \neq j \\ \frac{1}{p_j} & \text{si } h = j \end{cases}, \quad (3)$$

où $\{e_j / 1 \leq j \leq m\}$ est la base canonique de R^m .

La formule donnant le coefficient cherché repose sur la formule suivante de décomposition conformément à la partition $\{I_\ell / 1 \leq \ell \leq L\}$, de l'inertie totale du nuage :

$$\sum_{i \in I} p_i \cdot ||f_J^i - g_J||^2 = \sum_{1 \leq \ell \leq L} \left(\sum_{i \in I_\ell} p_i \cdot ||f_J^i - g_J^\ell||^2 \right) + \sum_{1 \leq \ell \leq L} p(I_\ell) ||g_J^\ell - g_J||^2, \quad (4)$$

dans cette formule : g_J désigne le centre de gravité du nuage $N(I)$ dont la suite des coordonnées est définie par $(p_j / j \in J)$, $p(I_\ell)$ est le poids de la ℓ -ème classe ; c'est-à-dire $\sum \{p_i / i \in I_\ell\}$, enfin g_J^ℓ indique le centre de gravité de la ℓ -ème classe dont la suite des coordonnées est définie par

$$\left\{ \left(\sum_{i \in I_\ell} f_{ij} \right) / \left(\sum_{i \in I_\ell} p_i \right) \right\} / 1 \leq j \leq m$$

La part d'inertie expliquée par la classification est définie par le rapport

$$\sum_{1 \leq \ell \leq L} p(I_\ell) ||g_J^\ell - g_J||^2 / \sum_{i \in I} p_i \cdot ||f_J^i - g_J||^2 \quad (5)$$

La contribution de la classe I à ce rapport est

$$p(I_\ell) ||g_J^\ell - g_J||^2 / \sum_{i \in I} p_i \cdot ||f_J^i - g_J||^2 \quad (6)$$

Dans ces conditions, le coefficient de responsabilité de la modalité j_0 dans la formation de la classe I_ℓ , est défini comme la contribution orientée de la composante j_0 au rapport (6).

La contribution (non orientée) de j_0 à (6) est définie par

$$\gamma(j_0 / I) = \frac{p(I_\ell) (f_{j_0}^\ell - p_{j_0})^2}{\sum_{i \in I} p_i (f_{j_0}^i - p_{j_0})^2} \quad (7)$$

où nous avons noté $f_{j_0}^\ell$ la proportion conditionnelle

$$f_{I j_0} / p(I_\ell) = \frac{\sum \{f_{i j_0} / i \in I_\ell\}}{\sum \{p_i / i \in I_\ell\}} \quad (8)$$

L'expression (7) ne permet de distinguer le sens de la contribution de la modalité j_0 dans la formation de la classe I_ℓ ; nous lui préférons la mesure orientée

$$C(j_0/I_\ell) = \frac{p(I_\ell)(f_{j_0}^\ell - p_{j_0})}{\sqrt{\sum_{i \in I} p_i (f_{j_0}^i - p_{j_0})^2}} \quad (9)$$

2. CALCULS EFFECTUES

L'expression de la formule (9) du coefficient, support des calculs, peut s'écrire en fonction du tableau (1) ci-dessus de départ, sous la forme suivante

$$C(j_0/I_\ell) = \frac{\sqrt{k_\ell} \left(\sum \{(k_{ij_0}/k_\ell)/i \in I_\ell\} - \sum \{(k_{ij_0}/k_{..})/i \in I\} \right)}{\sqrt{\sum_{i \in I} k_i \left[(k_{ij_0}/k_i) - (k_{j_0}/k_{..}) \right]^2}}, \quad (10)$$

avec des notations qu'on comprend :

$k_\ell = \sum \{k_i./i \in I_\ell\}$ où $k_i.$ a été précisé ci-dessus et où k_{j_0} a l'expression analogue suivante

$$k_{j_0} = \sum \{k_{ij_0}/i \in I\}$$

Successivement, pour chacun des tableaux de la forme $I \times J^{(h)}$, $1 \leq h \leq k$, le programme exécute les calculs suivants :

- pour chaque élément i de I , la somme $\sum \{k_{ij}/j \in J\}$ qu'on place dans le tableau $VVJ(I)$,

- pour chaque classe I_ℓ et pour chaque modalité j_0 de J , la somme $\sum \{k_{ij_0}/i \in I_\ell\}$ qu'on installe dans le tableau $VI(J)$,

- pour chaque modalité j_0 de J , la somme $\sum \{k_{ij_0}/i \in I\}$ qu'on situe dans le tableau $VVI(J)$,

- pour chaque classe I_ℓ , la somme $\sum \{k_{ij}/(i,j) \in I_\ell \times J\} = k_\ell$ qu'on trouvera dans le tableau $SOM(L)$,

- pour tout j_0 de J , la variance de la composante j_0 qu'on met dans le tableau $VAR(J)$,

- enfin, pour toute classe I_ℓ et toute modalité j_0 , le coefficient de discrimination de la classe I_ℓ par j_0 , lequel est chargé dans le tableau $C(L,J)$.

3. LES PARAMETRES DU PROGRAMME

Les données nécessaires pour le programme sont

- 1) Le cardinal de l'ensemble classifié donc ici celui des objets :
CARDI
- 2) Le cardinal de l'ensemble total des modalités ; c'est-à-dire, le total sur tous les tableaux du nombre de modalités concernant chacun, soit
CARDJ
- 3) Le nombre de tableaux : NCAR
- 4) Le nombre de classes d'objets : NBC
- 5) Le numéro d'ordre global par rapport à CARDJ de la dernière modalité de chaque tableau ; ces numéros sont rangés dans le tableau CARDIX à partir de l'indice 2. La première case repérant la colonne 0.
- 6) Le cardinal de chacune des classes d'objets par le tableau CARD(L)
- 7) Le tableau de données, lu sans distinction des tableaux composants, pour chaque objet sont lues les CARDJ modalités, ce pour tous les objets :
V(I,J).
- 8) Pour chaque classe L, le numéro d'ordre des éléments de I qui forment la classe L, dans le tableau IN(L,K) (il y a CARD(L) éléments par classe).

Notons que les formats 106, ainsi que 130-140-150-160 sont des formats relatifs à des titres de présentation globale pour le premier, et spécifiques aux modalités de chaque tableau pour les suivants. Le format 1100 sert à la lecture des données objet par objet.

Les dimensions de ces tableaux sont :

```
dim V = CARDI x CARDJ
dim CARDIX = NCAR + 1
dim CARD = NBC
dim IN = NBC x (Max {CARD(L), L=1, NBC}).
```

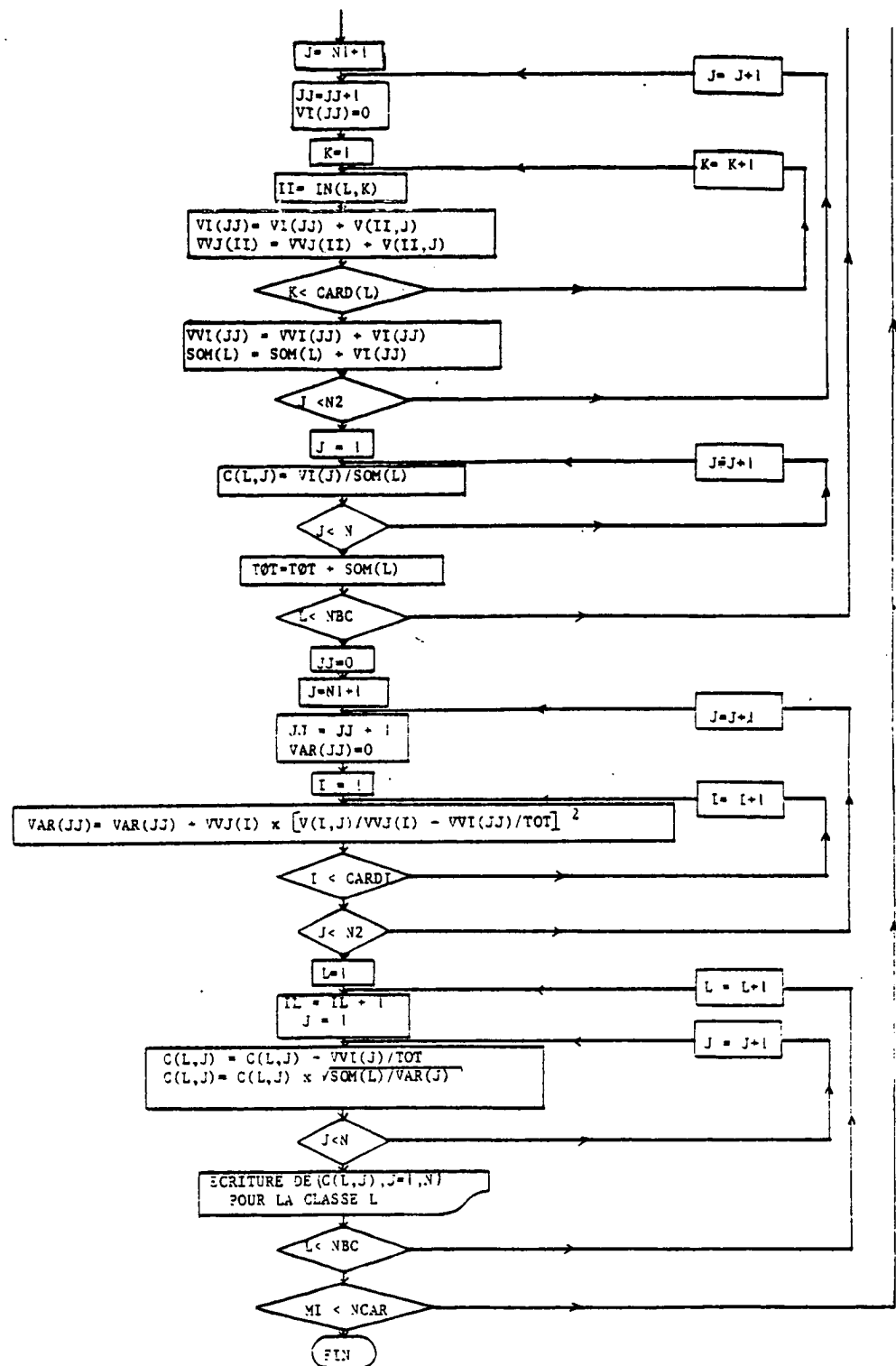
Seront également à préciser les dimensions des tableaux intermédiaires pour les calculs

```
dim VI = nombre maximal de modalités d'un tableau
dim VVI = "      "      "      "      "      "
dim VAR = "      "      "      "      "      "
dim C = NBCx"      "      "      "      "
dim VVJ = CARDI
dim SOM = NBC
```

4. L'ORGANISATION DES CALCULS DU PROGRAMME (*)

Après lecture des paramètres nécessaires, il y a le choix selon le tableau des modalités, des formats de présentation des sorties. Par différence des deux éléments successifs du tableau CARDIX est calculé le cardinal

(*) Voir organigramme ci-après.



5. LISTAGE.


```

        DIMENSION B(2150)
        CALL INIT (B(1),B(50),B)
        STOP
        END
        SUBROUTINE INIT (CARDI,CARD,B)
        INTEGER CARDI(1),CARD(1)
        DIMENSION B(1)
        INTEGER CARDI,CARDJ,NCAR,NBC
C
C C LECTURE DU NOMBRE D'OBJETS, CARDI
C C LECTURE DU NOMBRE DE VARIABLES, CARDJ
C C LECTURE DU NOMBRE DE TABLEAU, NCAR
C C LECTURE DU NOMBRE DE CLASSES D'OBJETS, NBC
C
        READ(41,100) CARDI,CARDJ,NCAR,NBC
C
C C LECTURE DU NUMERO DE LA DERNIERE VARIABLE DE CHAQUE TABLEAU
C C ,CARDI
C
        READ (41,100) (CARDI(I),I=2,NCAR+1)
        100 FORMAT(20I4)
C
C C LECTURE DU CARDINAL DE CHAQUE CLASSE D'OBJETS :CARD(L)
C
        READ(41,100) (CARD(L),L=1,NBC)
        CARDI(1)=0
        WRITE(20,100) (CARDI(I),I=1,NCAR+1)
        WRITE(20,100) (CARD(L),L=1,NBC)
        MAXC=0
        DO 1 K=1,NBC
        IF(MAXC.LT.CARD(K)) MAXC=CARD(K)
        1 CONTINUE
        MAXM=0
        DO 2 J=2,NCAR+1
        N1=CARDI(J)-CARDI(J-1)
        IF(MAXM.LT.N1) MAXM=N1
        2 CONTINUE
        I1=NCAR+2
        I2=I1+NBC
        I3=I2+CARDI*CARDJ
        I4=I3+CARDI
        I5=I4+NBC
        I6=I5+MAXM
        I7=I6+MAXM
        I8=I7+MAXM
        I9=I8+NBC*MAXM
        I10=I9+NBC*MAXC
        CALL COEF(CARDI,CARDJ,NCAR,NBC,B(1),B(I1),B(I2),B(I3),
        B(I4),B(I5),B(I6),B(I7),B(I8),B(I9),B(I10))
        RETURN
        END
        SUBROUTINE COEF(CARDI,CARDJ,NCAR,NBC,CARDI,CARD,V,VVJ,SOM,VI,VVI,
        1 VAR,C,IN,FMT)
C
C C DEGRE DE DISCRIMINATION DES CLASSES OBTENUES PAR CHI2-COR
C
        INTEGER CARDI(1),CARD(1),IN(NBC,1)
        REAL V(CARDI,1),VI(1),VVI(1),VVJ(1),SOM(1),VAR(1),C(NBC,1)
        INTEGER FMT(1)
        INTEGER CARDI,CARDJ,NCAR,NBC
        REWIND 20
        READ(20,100) (CARDI(I),I=1,NCAR+1)
        READ(20,100) (CARD(L),L=1,NBC)
C
C C LECTURE DES NUMEROS DES ITEMS DE LA CLASSE L : IN(L,K)
C
        DO 10 L=1,NBC
        READ(41,100) (IN(L,K),K=1,CARD(L))
        10 CONTINUE
C
C C FORMAT DE LECTURE DES DONNEES
C
        READ(41,100) NC
        READ(41,1000) (FMT(J),J=1,20*NC)
C
C C LECTURE DU TABLEAU DE DONNEES.
C
        DO 1 I=1,CARDI
        READ(10,FMT) (V(I,J),J=1,CARDJ)
        1 CONTINUE
        CALL FORM(FMT)
        DO 15 MI=1,NCAR
        IL=0
        WRITE(42,110) MI
        110 FORMAT(///,2X,'TABLEAU',I3,/)
        CALL FORM(FMT)
        CALL FORM(FMT)
        N1=CARDI(MI)
        N2=CARDI(MI+1)
        N=N2-N1
        TOT=0
        DO 4 I=1,CARDI
        VVJ(I)=0
        4 CONTINUE
        DO 5 JJ=1,N
        VVI(JJ)=0
        5 CONTINUE
C

```

```

C  CALCUL DE LA SOMME SUR J POUR CHAQUE OBJET: VVJ(I)
CCC CALCUL DE LA SOMME SUR I DE LA CLASSE POUR CHAQUE MODALITE, VI(J)
CCC CALCUL DE LA SOMME SUR TOUT I POUR CHAQUE MODALITE: VVI(J)
C  CALCUL DE LA SOMME SUR J ET SUR I DE LA CLASSE L: SOM(L)
DO 9 L=1,NBC
  JJ=0
  SOM(L)=0.
  DO 7 J=N1+1,N2
    JJ=JJ+1
    VI(JJ)=0
    DO 6 K=1,CARD(L)
      II=IN(L,K)
      VI(JJ)=VI(JJ)+V(II,J)
      VVJ(II)=VVJ(II)+V(II,J)
6  CONTINUE
  VVI(JJ)=VVI(JJ)+VI(JJ)
  SOM(L)=SOM(L)+VI(JJ)
7  CONTINUE
  DO 8 J=1,N
    C(L,J)=VI(J)/SOM(L)
8  CONTINUE
  TOT=TOT+SOM(L)
9  CONTINUE
  JJ=0

CCC CALCUL DE LA VARIANCE GLOBALE DE JEME MODALITE DU TABLEAU MI
DO 12 J=N1+1,N2
  JJ=JJ+1
  VAR(JJ)=0.
  DO 11 I=1,CARDI
    VAR(JJ)=VAR(JJ)+VVJ(I)*(V(I,J)/VVJ(I)-VVI(JJ)/TOT)**2
11 CONTINUE
12 CONTINUE

CCC CALCUL DU COEF DE DISCRIMINATION DE LA CLASSE D'OBJETS L
PAR LA MODALITE J: C(L,J)
DO 14 L=1,NBC
  IL=IL+1
  DO 13 J=1,N
    C(L,J)=C(L,J)-VVI(J)/TOT
    C(L,J)=C(L,J)*SQRT(SOM(L)/VAR(J))
13 CONTINUE
  WRITE (42,120) IL,(C(L,J),J=1,N)
120 FORMAT(1X,'#CLASSE',I3,'#',17(F6.3,'#'))
14 CONTINUE
15 CONTINUE
100 FORMAT(20I4)
1000 FORMAT(20A4)
  RETURN

END
SUBROUTINE FORM(FMT)
  INTEGER FMT(1)
  READ(41,100)NC
  READ(41,1000) (FMT(J),J=1,20*NC)
  WRITE(42,1100) (FMT(J),J=1,20*NC)
100 FORMAT(20I4)
1000 FORMAT (20A4)
1100 FORMAT (33A4)
  RETURN
END

```

III - CROISEMENT DE CLASSIFICATIONS "FLOUES"

PROGRAMME CROISFLOU (M. HARDOUIN)

1. FONCTION GENERALE DU PROGRAMME

S et C sont deux ensembles disjoints d'attributs de description d'un même ensemble E d'individus ou objets. S et C sont généralement relatifs à deux aspects sémantiques différents d'une même étude. Ainsi, dans les questionnaires supportant les enquêtes à caractère psycho-sociologique, on peut distinguer deux parties : la première est formée de questions permettant une description purement sociologique du sujet (par exemple son âge, sa catégorie socio-professionnelle, ses activités, son lieu de résidence, etc...) et la deuxième, qui a un caractère psychologique, décrit la réaction du sujet à des questions de nature comportementale où l'estimation subjective de l'individu peut intervenir.

On considère deux partitions de S et C :

$$\pi(S) = \{S_r / 1 \leq r \leq h\}$$

et

$$\pi(C) = \{C_s / 1 \leq s \leq k\}$$

qui correspondent généralement à deux classifications polythétiques par proximité de S et de C. $\pi(S)$ (resp. $\pi(C)$) peut par exemple être définie à partir d'un niveau "significatif" de l'arbre hiérarchique des classifications sur S (resp. C).

$\pi(S)$ (resp. $\pi(C)$) définit une "typologie" de l'ensemble E des individus où, relativement à une même classe d'attributs S_r ou C_s ($1 \leq r \leq h$, $1 \leq s \leq k$) définissant un "type", on mesure au moyen d'une proportion le "degré d'appartenance" d'un individu donné au type. Ce degré d'appartenance se réduit à l'une des deux valeurs 0 ou 1 dans le cas où la classe d'attributs contient un seul élément descriptif.

Ce programme établit la table de croisement entre $\pi(S)$ et $\pi(C)$, où à l'intersection de la ligne r et de la colonne s ($1 \leq r \leq h$, $1 \leq s \leq k$) on trouve la valeur d'un indice mesurant le lien entre le profil d'attitude défini par la classe S_r et celui, défini par la classe C_s ($1 \leq r \leq h$, $1 \leq s \leq k$). Cet indice généralise celui entre deux attributs et est, dans le sens suivant, conforme à la statistique du χ^2 : Si les différentes classes S_r ($1 \leq r \leq h$) (resp. C_s ($1 \leq s \leq k$)) se réduisent chacune à un seul attribut associé à l'une des h (resp. k) modalités d'une variable qualitative nominale, l'indice α_{rs} , à l'intersection de la ligne r et de la colonne s n'est autre que la contribution orientée à la statistique du χ^2 du tableau de contingence ainsi défini ; $\chi^2 = \sum_{rs} \{\alpha_{rs}^2 / 1 \leq r \leq h, 1 \leq s \leq k\}$.

Un cas fréquent de l'utilisation du programme est intermédiaire entre la première et la dernière situations envisagées, c'est celui où chaque S_r ($1 \leq r \leq h$) représente exactement une modalité d'un caractère descriptif exogène à h modalités et où les classes C_s ($1 \leq s \leq k$) sont obtenues à partir d'une classification polythétique par proximité d'un ensemble C d'attributs orientés (si $a \in C \Rightarrow \bar{a} \notin C$) ; de la sorte, chaque classe C_s définit un profil de comportement.

L'étude théorique complète qui soustend ce programme se trouve au chapitre 3 de l'ouvrage "Classification et analyse ordinaire des données".

2. CALCULS EFFECTUES

La donnée globale est sous la forme d'un tableau d'incidence de dimension $CARDI \times CARDJ$ où $CARDI$ (resp. $CARDJ$) désigne le cardinal de l'ensemble des individus (resp. des variables). On désignera par $J1$ (resp. $J2$) le cardinal du sous ensemble des attributs de la première espèce jouant le rôle de S ci-dessus (resp. de la deuxième espèce jouant le rôle de C ci-dessus). On suppose qu'on a procédé à deux classifications séparées de l'ensemble des $J1$ attributs d'une part (à partir d'un tableau d'incidence de dimension $CARDI \times J1$) et de l'ensemble des $J2$ attributs d'autre part (à partir d'un tableau d'incidence de dimension $CARDI \times J2$). $NBCL1$ (resp. $NBCL2$) est le nombre de classes de la première (resp. seconde) classification ; d'autre part, pour continuer à se conformer aux notations du programme, nous désignerons ici par J (resp. K) ci-dessus noté r (resp. s) l'indice d'une classe courante de la première (resp. seconde) classification.

$$\phi_J(i) = \frac{1}{B(J)} \sum \{T(i,a)/a \leftarrow J\text{-ème classe}\}$$

$$\phi_K(i) = \frac{1}{B(K)} \sum \{T(i,a)/a \leftarrow K\text{-ème classe}\}$$

où $T(i,a)$ ($=0$ ou 1) est l'élément de la ligne i et de la colonne indexée par a du tableau d'incidence et où $B(J)$ (resp. $B(K)$) est le cardinal de la J -ème (resp. K -ème) classe, $1 \leq i \leq CARDI$, $1 \leq a \leq CARDJ$, $1 \leq J \leq NBCL1$ et $1 \leq K \leq NBCL2$.

L'indice brut d'association entre la J -ème classe de la première classification et la K -ème classe de la seconde classification est défini par

$$v_{JK} = \sum \{\phi_J(i)\phi_K(i)/1 \leq i \leq CARDI\}$$

D'autre part

$$v_{J.} = \sum \{v_{JK}/1 \leq K \leq NBCL2\}$$

et
$$v_{.K} = \sum \{v_{JK}/1 \leq J \leq NBCL1\}$$

définissent respectivement l'"importance numérique" de la J -ème et de la K -ème classe ($1 \leq J \leq NBCL1$, $1 \leq K \leq NBCL2$).

On peut remarquer que dans le cas où $B(J) = 1$ pour tout $J=1,2,\dots,NBCL1$ où $B(K)=1$ pour tout $K=1,2,\dots,NBCL2$ et où la réunion des attributs que sépare une même classification (à raison d'un attribut par classe) représente l'ensemble des modalités d'un même caractère descriptif, la table des nombres v_{JK} définit un tableau de contingence ordinaire.

L'indice "centré réduit" d'association entre la J -ème classe de la première classification et la K -ème classe de la seconde classification, est défini par

$$t_{JK} = \frac{v_{JK} - (v_{J.} v_{.K} / v_{..})}{\sqrt{(v_{J.} v_{.K} / v_{..})}},$$

$1 \leq J \leq NBCL1$, $1 \leq K \leq NBCL2$.

3. PARAMETRES ET ORGANISATION DU PROGRAMME

On suppose déjà établies deux partitions sur deux ensembles disjoints d'attributs relativement à la description d'un même ensemble d'individus.

La lecture des données ainsi que tous les calculs sont effectués individu par individu ; c'est-à-dire, ligne par ligne, relativement à une organisation du tableau des données Individus x Variables. L'espace mémoire demandé n'est donc pas influencé par le nombre d'individus.

CARDI désigne le cardinal de l'ensemble des individus ; c'est-à-dire le nombre de lignes du tableau d'incidence des données.

CARDJ désigne le cardinal de l'ensemble des attributs ; lequel étant supposé scindé en deux sous-ensembles complémentaires qui, comme nous venons de le mentionner, ont été au préalable classifiés.

NBCL1 est le nombre de classes de la première classification et NBCL2 est le nombre de classes de la deuxième classification.

L'exécution du programme nécessite la création des tableaux suivants:

A1, A2 et T : tableaux à une ligne de dimension CARDJ.

B1, SOM1, SSOM1 et TN1 : tableaux à une ligne de dimension NBCL1.

B2, SOM2, SSOM2 et TN2 : tableaux à une ligne de dimension NBCL2.

TN et TCR : tableaux à double entrée de dimension NBCL1 x NBCL2.

Le tableau A₁ (resp. A₂) contient les adresses des attributs intervenant dans la première (resp. seconde) classification ; repérées de la façon suivante :

$A1(j)=0 \Leftrightarrow$ la j-ème variable n'a pas été retenue dans le premier ensemble classifié.

$A1(j)=r \Leftrightarrow$ la j-ème variable fait partie de la r-ème classe de la première classification,

$1 \leq j \leq \text{CARDJ}$ et $1 \leq r \leq \text{NBCL1}$.

De même

$A2(j)=0 \Leftrightarrow$ la j-ème variable n'a pas été retenue dans le deuxième ensemble classifié.

$A2(j)=s \Leftrightarrow$ la j-ème variable fait partie de la s-ème classe de la deuxième classification,

$1 \leq j \leq \text{CARDJ}$ et $1 \leq s \leq \text{NBCL2}$.

T contient une ligne courante du tableau des données.

B1 (resp. B2) contient les cardinaux des classes de la première (resp. seconde) classification :

B1(J) est le cardinal de la J-ème classe de la première classification,

$1 \leq J \leq \text{NBCL1}$.

$B2(K)$ est le cardinal de la K -ème classe de la deuxième classification, $1 \leq K \leq \text{NBCL2}$.

A la lecture d'une ligne du tableau d'incidence, chargée dans T , on remplit les tableaux $SOM1$ et $SOM2$ conformément aux formules

$$SOM1(J) = \sum \{T(i,a)/a \leftarrow J\text{-ème classe}\}$$

$$SOM2(K) = \sum \{T(i,a)/a \leftarrow K\text{-ème classe}\}$$

$1 \leq J \leq \text{NBCL1}$, $1 \leq K \leq \text{NBCL2}$.

Ainsi, à chaque étape, $SOM1(J)$ (resp. $SOM2(K)$) contient pour le i -ème individu, le nombre d'attributs de la J -ème classe de la première classification (resp. K -ème classe de la seconde classification) possédés par l'individu i .

A la fin du déroulement du fichier, on aura calculé

$$SSOM1(J) = \sum \{SOM1(J)/1 \leq i \leq \text{CARDI}\}$$

$$SSOM2(K) = \sum \{SOM2(K)/1 \leq i \leq \text{CARDI}\},$$

($1 \leq J \leq \text{NBCL1}$, $1 \leq K \leq \text{NBCL2}$) qui représentent respectivement les "poids" numériques de la J -ème classe de la première classification et de la K -ème classe de la seconde classification.

C'est ligne par ligne qu'on constitue les produits

$$SOM1(J) \times SOM2(K)$$

dont la somme finale fournit $TN(J,K)$:

$$TN(J,K) = \frac{1}{B1(J)B2(K)} \sum \{SOM1(J)SOM2(K)/1 \leq i \leq \text{CARDI}\}$$

qui est le reflet de ce que nous avons noté v_{JK} au paragraphe précédent.

D'autre part, $TN1(J)$ et $TN2(K)$ représentent respectivement $v_{J.}$ et $v_{.K}$. Enfin, TNN représente $v_{..}$. Le tableau des nombres t_{JK} (cf. § 2 ci-dessus) est noté dans le programme $TCR(J,K)$, $1 \leq J \leq \text{NBCL1}$, $1 \leq K \leq \text{NBCL2}$.

4. USAGE ET SORTIES DU PROGRAMME

a) Pour utiliser ce programme, on doit :

- remplacer 2 cartes de dimensionnement (celles de numéros 3 et 4) ainsi que la carte de format des données lue sous l'étiquette `FORMAT 40` en respectant les tailles des tableaux donnés précédemment.

- écrire 5 cartes au moins de données :

1ère carte : `CARDI`, `CARDJ`, `NBCL1`, `NBCL2` où ces quatre nombres entiers sont écrits en format `4I4`.

2ème carte et les suivantes si elle ne suffit pas : `A1(J)`, $J=1$, `CARDJ`, sous le format `40I2`.

3ème carte et les suivantes si elle ne suffit pas : A2(J), J=1, CARDJ, sous le format 40I2.

Ainsi, par exemple, une analyse sur un ensemble de 110 attributs au total (CARDJ=110) nécessitera 3 cartes pour la définition de A1 et 3 cartes pour la définition de A2. D'autre part, avec ce format, le nombre maximum de classes admis pour chacune des deux classifications à croiser est de 99.

4ème carte et si besoin, les suivantes : B1(J), J=1, NBCL1, sous le format 40I2.

5ème carte et si besoin, les suivantes : B2(J), J=1, NBCL2, sous le format 40I2.

Avec un tel format, une même classe de l'une des deux classifications peut contenir jusqu'à 99 attributs.

Ensuite intervient le paquet des données, si ces dernières sont sur cartes. Autrement, il convient de modifier l'instruction de lecture en remplaçant 105 par le numéro d'étiquette logique à laquelle a été assigné le fichier d'entrée (disque ou bande).

La dénomination "première classification" sera donnée à celle des deux qui correspond aux variables identifiées par la carte A1. On peut choisir arbitrairement l'une des deux ; le programme n'est limité en écriture dans aucune des deux dimensions car on a utilisé un format variable. Cependant, pour une meilleure présentation des tableaux, il est conseillé d'appeler "première classification", celle qui a le plus grand nombre de classes ; la largeur du tableau imprimé (TN ou TCR) étant de 11 colonnes. Si la deuxième classification comporte plus de 11 classes, ces tableaux sont découpés par tranches de 11 colonnes écrites à la suite les unes des autres, comme dans l'exemple donné en annexe.

b) Sorties :

Le programme recopie les cartes A1(resp. A2) précédées du titre

"Adresse des classes retenues dans la première (resp. deuxième) classification".

De même, il recopie les cartes B1(resp. B2) avec le titre

"Cardinaux des classes de la première (resp. deuxième) classification"

Puis, sous le titre

"Poids des classes de la première (resp. deuxième) classification",

le tableau à une ligne des nombres

$SSOM1(J)/(B1(J) \times CARDI), 1 \leq J \leq NBCL1$
(resp. $SSOM2(K)/(B2(K) \times CARDI), 1 \leq K \leq NBCL2$).

Puis, sous le titre

"Importance numérique des classes de la première (resp. deuxième) classification",

le tableau des valeurs

$TN1(J), 1 \leq J \leq NBCL1$ (resp. $TN2(K), 1 \leq K \leq NBCL2$).

Ensuite, sous le titre

"Mesure des associations entre les classes du croisement", le tableau à double entrée des nombres

$TN(J,K), 1 \leq J \leq NBCL1, 1 \leq K \leq NBCL2$

Enfin, sous le titre

"Mesure des associations centrées réduites entre les classes" le tableau à double entrée des nombres

$TCR(J,K), 1 \leq J \leq NBCL1, 1 \leq K \leq NBCL2$.

5. SUR L'INTERPRETATION DES RESULTATS.

Nous avons déjà ci-dessus signalé (cf. §1) qu'un des usages les plus courants de ce programme est le croisement des différentes modalités (dont chacune définit une classe "nette" d'individus) d'un caractère descriptif exogène avec les différentes classes d'attributs (dont chacune représente une classe "floue" de sujets) issue d'une classification par proximité d'un ensemble C d'attributs de comportement.

Une situation intéressante de ce dernier type de croisement est définie lorsque la classification "nette" sur l'ensemble des individus est précisément obtenue au moyen d'une classification polythétique par proximité de l'ensemble E des individus dans leur description à travers l'ensemble C des attributs. Un tel croisement permet de préciser quelles sont les classes d'attributs C_r auxquelles se réfère une même classe E_s et inversement, quelles sont les classes d'individus E_s que touche une même classe C_r d'attributs.

Revenons au cas le plus général pour insister sur l'importance qu'il y a à accompagner le tableau des nombres t_{JK} représentant les mesures d'association centrées réduites par celui des nombres v_{JK} représentant les mesures brutes, $1 \leq J \leq NBCL1, 1 \leq K \leq NBCL2$. En effet, relativement à un même individu i , une valeur relativement élevée (resp. basse) du produit $\phi_J(i)\phi_K(i)$ indique une association relativement "positive" (resp. "négative") des deux classes d'attributs J et K sur l'individu i , $1 \leq i \leq CARDI$. Le nombre $v_{JK} = \sum \{\phi_J(i)\phi_K(i) / 1 \leq i \leq CARDI\}$ permet dans ces conditions d'évaluer par rapport à $v_J, v_K/v_{..}$ et de façon globale, le caractère positif ou négatif de l'association entre les deux classes d'attributs J et K, $1 \leq J \leq NBCL1, 1 \leq K \leq NBCL2$.

Enfin, le programme édite les tableaux (à une ligne) des nombres

$POIDS1(J) = SSOM1(J) / (B1(J) \times CARDI), 1 \leq J \leq NBCL1,$

$POIDS2(K) = SSOM2(K) / (B2(K) \times CARDI), 1 \leq K \leq NBCL2,$

où $POIDS1(J)$ (resp. $POIDS2(K)$) représente la présence "moyenne" d'un attribut de la classe J (resp. K) sur l'ensemble des individus. Ce nombre est in-

dépendant de la deuxième (resp. première) classification. Alors que v_J (resp. v_K) qui représente l'importance numérique du type défini par la J-ème (resp. K-ème) classe de la première (resp. seconde) classification, dépend de la seconde (resp. première) classification.

6. LISTAGE.

```

C *****
C CROISFLOU
C
C CE PROGRAMME CROISE DEUX CLASSIFICATIONS FLOUES D'ATTRIBUTS
C DESCRIPTIFS. IL NECESSITE LA MODIFICATION DES DEUX LIGNES DE
C DIMENSION, EN CREANT LES TABLEUX:
C A1,A2,T DE DIMENSION CARDJ
C B1,SOM1,SSOM1,TN1, DE DIMENSION NBCL1
C B2,SOM2,SSOM2,TN2, DE DIMENSION NBCL2
C TN ET TCR DE DIMENSION (NBCL1,NBCL2)
C A1= ADRESSE DES VARIABLES DE LA PREMIERE CLASSIFICATION EN I2
C A1(J)=R SIGNIFIE LA J IEME VARIABLE FAIT PARTIE DE LA R IEME
C CLASSE DE LA PREMIERE CLASSIFICATION, SINON A1(J)=0
C A2= ADRESSE DES VARIABLES DE LA DEUXIEME CLASSIFICATION (ID A1)
C B1=CARDINAUX DES CLASSES DE LA PREMIERE CLASSIFICATION,EN I2
C B2= CARDINAUX DES CLASSES DE LA DEUXIEME CLASSIFICATION,EN I2
C T CONTIENT LES DONNEES QUI SONT LUES DANS LE FICHIER IENT,SOUS
C LE FORMAT NFORM
C LE PROGRAMME LIT LES PARAMETRES SUR L'UNITE 15 ET ECRIT LES
C RESULTATS SUR L'UNITE 16
C *****
C
C INTEGER T,CARDI,CARDJ,A1,A2,B1,B2,SOM1,SOM2,NFORM(20)
C DIMENSION A1(97),A2(97),B1(5),B2(12),T(97),SOM1(5),SOM2(12)
C DIMENSION TN(15,12),TN1(5),TN2(12),TCR(15,12),SSOM1(5),SSOM2(12)
C INTEGER FORM1(9),FORM2(9),FORM3(9),FORM4(14),FORM5(12),FORM(10)
C DATA FORM/1,2,3,4,5,6,7,8,9,10/
C DATA FORM1/'1X',' ','9HC','ARDC','L. N','I2','1H','=','I2,2','X
C 1)'/
C DATA FORM2/'1X',' ','7HP','OIDS',' N,I','2,1H','=,F4','2,2','X
C 1)'/
C DATA FORM3/'1X',' ','7HI','MPNU',' N,I','2,1H','=,F4','2,2','X
C 1)'/
C DATA FORM4/'11X',' ','8H','CLAS','2 N','I2,1','X),1',
C 1'H:1','1X','11('','1H','))'/
C DATA FORM5/'1X','7HCL','AS1','N,I2',' ','1H:','1X','F8.4
C 1','1X','1H:'))'/
C READ(15,1) CARDI,CARDJ,NBCL1,NBCL2,IENT
C READ(15,3) NFORM
C READ(15,2)(A1(J),J=1,CARDJ)
C WRITE(16,12) (A1(J),J=1,CARDJ)
C READ(15,2)(A2(J),J=1,CARDJ)
C WRITE(16,13) (A2(J),J=1,CARDJ)
C READ(15,2)(B1(J),J=1,NBCL1)
C NBLIGN=NBCL1/8
C NBR=NBCL1-8*NBLIGN
C WRITE(16,24)
C IF (NBLIGN.EQ.0) GO TO 96
C WRITE(16,14)(J,B1(J),J=1,8*NBLIGN)
C 96 IF(NBR.EQ.0)GO TO 98
C FORM1(2)=FORM(NBR)
C WRITE(16,FORM1)(J,B1(J),J=8*NBLIGN+1,NBCL1)
C 98 CONTINUE
C READ(15,2)(B2(J),J=1,NBCL2)
C NCLIGN=NBCL2/8
C NCR=NBCL2-8*NCLIGN
C WRITE(16,25)
C IF (NCLIGN.EQ.0) GO TO 97
C WRITE(16,14) (J,B2(J),J=1,8*NCLIGN)
C 97 IF (NCR.EQ.0) GO TO 99
C FORM1(2)=FORM(NCR)
C WRITE (16,FORM1)(J,B2(J),J=8*NCLIGN+1,NBCL2)
C 99 CONTINUE
C DO 31 J=1,NBCL1
C DO 31 K=1,NBCL2
C 31 TN(J,K)=0
C DO 32 J=1,NBCL1
C SSOM1(J)=0
C 32 TN1(J)=0
C DO 33 K=1,NBCL2
C SSOM2(K)=0
C 33 TN2(K)=0

```

```

I=1
4 READ(IENT,NFORM)(T(J),J=1,CARDJ)
DO 21 K=1,NBCL1
21 SOM1(K)=0
DO 22 K=1,NBCL2
22 SOM2(K)=0
J=1
7 CONTINUE
DO 5 K=1,NBCL1
IF (A1(J).EQ.K) SOM1(K)=SOM1(K)+T(J)
5 CONTINUE
DO 6 K=1,NBCL2
IF (A2(J).EQ.K) SOM2(K)=SOM2(K)+T(J)
6 CONTINUE
J=J+1
IF (J.LE.CARDJ) GO TO 7
DO 41 J=1,NBCL1
DO 41 K=1,NBCL2
41 TN(J,K)=TN(J,K)+(SOM1(J)*SOM2(K))
DO 42 J=1,NBCL1
42 SSOM1(J)=SSOM1(J)+SOM1(J)
DO 47 K=1,NBCL2
47 SSOM2(K)=SSOM2(K)+SOM2(K)
I=I+1
IF (I.LE.CARDI) GO TO 4
DO 43 J=1,NBCL1
DO 43 K=1,NBCL2
TN(J,K)=TN(J,K)/(B1(J)*B2(K))
43 TN1(J)=TN1(J)+TN(J,K)
DO 44 K=1,NBCL2
DO 44 J=1,NBCL1
44 TN2(K)=TN2(K)+TN(J,K)
TNN=0
DO 45 J=1,NBCL1
45 TNN=TNN+TN1(J)
DO 46 J=1,NBCL1
DO 46 K=1,NBCL2
IF ((TN1(J)*TN2(K)).EQ.0) TCR(J,K)=0
IF ((TN1(J)*TN2(K)).NE.0) TCR(J,K)=(TN(J,K)-(TN1(J)*TN2(K)/TNN))/
1SQRT(TN1(J)*TN2(K)/TNN)
46 CONTINUE
DO 48 J=1,NBCL1
48 SSOM1(J)=SSOM1(J)/(B1(J)*CARDI)
WRITE(16,26)
IF (NBLIGN.EQ.0) GO TO 56
WRITE(16,16) (J,SSOM1(J),J=1,8*NBLIGN)
56 IF (NBR.EQ.0) GO TO 58
FORM2(2)=FORM(NBR)
WRITE(16,FORM2)(J,SSOM1(J),J=8*NBLIGN+1,NBCL1)
58 CONTINUE
DO 49 J=1,NBCL2
49 SSOM2(J)=SSOM2(J)/(B2(J)*CARDI)
WRITE(16,27)
IF (NCLIGN.EQ.0) GO TO 57
WRITE(16,16) (J,SSOM2(J),J=1,8*NCLIGN)
57 IF (NCR.EQ.0) GO TO 59
FORM2(2)=FORM(NCR)
WRITE(16,FORM2)(J,SSOM2(J),J=8*NCLIGN+1,NBCL2)
59 CONTINUE
WRITE(16,28)
IF (NBLIGN.EQ.0) GO TO 66
WRITE(16,18) (J,TN1(J),J=1,8*NBLIGN)
66 IF (NBR.EQ.0) GO TO 68
FORM3(2)=FORM(NBR)
WRITE(16,FORM3) (J,TN1(J),J=8*NBLIGN+1,NBCL1)
68 CONTINUE
WRITE(16,29)
IF (NCLIGN.EQ.0) GO TO 67
WRITE(16,18) (J,TN2(J),J=1,8*NCLIGN)
67 IF (NCR.EQ.0) GO TO 69
FORM3(2)=FORM(NCR)
WRITE(16,FORM3)(J,TN2(J),J=8*NCLIGN+1,NBCL2)
69 CONTINUE
WRITE(16,100)
NDLIGN=NBCL2/11

```

```

NDR=NBCL2-11*NDLIGN
IF (NDLIGN.EQ.0) GO TO 82
DO 80 N=1,NDLIGN
WRITE(16,200) (K,K=11*N-10,11*N)
DO 81 J=1,NBCL1
WRITE (16,300) J,(TN(J,K),K=11*N-10,11*N)
81 CONTINUE
80 CONTINUE
82 IF (NDR.EQ.0) GO TO 83
FORM4(3)=FORM(NDR)
FORM4(11)=FORM(NDR)
WRITE (16,FORM4) (K,K=11*NDLIGN+1,NBCL2)
DO 84 J=1,NBCL1
FORM5(6)=FORM(NDR)
WRITE (16,FORM5) J,(TN(J,K),K=11*NDLIGN+1,NBCL2)
84 CONTINUE
83 CONTINUE
WRITE(16,101)
IF (NDLIGN.EQ.0) GO TO 72
DO 70 N=1,NDLIGN
WRITE(16,200) (K,K=11*N-10,11*N)
DO 60 J=1,NBCL1
WRITE (16,300) J,(TCR(J,K),K=11*N-10,11*N)
60 CONTINUE
70 CONTINUE
72 IF (NDR.EQ.0) GO TO 73
WRITE (16,FORM4) (K,K=11*NDLIGN+1,NBCL2)
DO 74 J=1,NBCL1
WRITE (16,FORM5) J,(TCR(J,K),K=11*NDLIGN+1,NBCL2)
74 CONTINUE
73 CONTINUE
1 FORMAT(5I4)
2 FORMAT(40I2)
3 FORMAT (20A4)
12 FORMAT (30X,'ADRESSES DES CLASSES RETENUES DANS LA PREMIERE CLASSI-
FICATION',30X,61(' '),1X,(65I2))
13 FORMAT (///30X,'ADRESSES DES CLASSES RETENUES DANS LA DEUXIEME CLA-
SSIFICATION',30X,61(' '),1X,(65I2))
24 FORMAT(///30X,'CARDINAUX DES CLASSES DE LA PREMIERE CLASSIFICATION
1',30X,51(' '))
14 FORMAT(1X,8('CARDCL. N',I2,'=',I2,2X))
25 FORMAT(///30X,'CARDINAUX DES CLASSES DE LA DEUXIEME CLASSIFICATION
1',30X,51(' '))
26 FORMAT (///30X,'POIDS DES CLASSES DE LA PREMIERE CLASSIFICATION',
130X,47(' '))
16 FORMAT(1X,8('POIDS N',I2,'=',F4.2,2X))
27 FORMAT (///30X,'POIDS DES CLASSES DE LA DEUXIEME CLASSIFICATION',
130X,47(' '))
28 FORMAT (///30X,'IMPORTANCE NUMERIQUE DES CLASSES DE LA PREMIERE CL
ASSIFICATION',30X,62(' '))
18 FORMAT(1X,8('IMPNU N',I2,'=',F4.2,2X))
29 FORMAT (///30X,'IMPORTANCE NUMERIQUE DES CLASSES DE LA DEUXIEME CL
ASSIFICATION',30X,62(' '))
100 FORMAT(///30X,'MESURE DES ASSOCIATIONS ENTRE LES CLASSES DU CROISE
MENT',30X,55(' '))
101 FORMAT(///30X,'MESURE DES ASSOCIATIONS CENTREES REDUITES ENTRE LES
CLASSES',30X,59(' '))
200 FORMAT(11X,11(' CLAS2 N',I2,1X), ' ',11X,11(11(' ')))
300 FORMAT (1X,'CLAS1 N',I2,11(' ',1X,F8.4,1X), ' ')
STOP
END

```

IV - PROGRAMME DE CONSTRUCTION DES ECHELLES

C. RISO-LEVY

1 - INTRODUCTION

Nous nous proposons d'exprimer ici le programme qui développe dans son expression la plus générale, l'algorithme de recherche des échelles optimales d'attitude (cf. ch. 9 § III.2)*; où, rappelons-le en termes intuitifs, il s'agit d'organiser m échelles a^1, a^2, \dots, a^m , présentant respectivement $r_1 + 1, r_2 + 1, \dots, r_m + 1$ modalités et se référant à une même "dimension sous-jacente", en une échelle "fine" d'attitude à $(r_1 + r_2 + \dots + r_m)$ modalités. La seule contrainte dans l'extension de l'algorithme sera liée à la capacité de la mémoire de l'ordinateur ; les limitations considérées dans ce programme sont :

$$m \leq 10, \quad s = \sum_j (r_j + 1) \leq 100 \quad \text{et} \quad t = \prod_j r_j \leq 20.000 ;$$

s est la somme des nombres de modalités des différents items et t est le nombre de sommets du parallélotope de représentation.

Une des difficultés importantes de l'algorithme étendu à m dimensions réside dans la détermination de toutes les échelles également optimales ; la présence de plus d'une échelle optimale peut être compatible avec l'hypothèse d'unidimensionnalité si certaines modalités des diverses échelles à composer, sont trop "proches", en tenant compte des possibilités d'erreurs dans la réponse des sujets.

Ce n'est que partiellement que la difficulté a été résolue en retenant, dans la recherche d'une même échelle, le dernier sommet dont l'indicateur de précedence i comporte plus d'une composante égale à 1 et en y revenant jusqu'à exploiter tous les chemins finissant en l'origine du treillis, pouvant achever une échelle optimale. Mais le principe de notre algorithme reste valable pour procéder à une recherche exhaustive (cf. § 4 ci-dessous).

L'auteur du programme étudié ici est C. Riso - Lévi dont d'ailleurs nous reprenons ici le texte d'analyse qu'il a établi à la suite de son travail. Le plan de ce grand paragraphe sera conforme aux précédents :

- § 2 - Calculs effectués
- § 3 - Paramètres et données du programme
- § 4 - Organisation et algorithmes du programme
- § 5 - Description des sorties
- § 6 - Liste des instructions du programme

2 - CALCULS EFFECTUES

Nous chercherons à utiliser ici les notations les plus proches de celles du programme sauf lorsque ces dernières deviennent par trop lourdes.

A chaque sommet IX du parallélotope de représentation se trouve affecté un entier positif ou nul n (IX) : fréquence absolue du "patron" représenté par IX. On distinguera trois étapes dans les calculs effectués.

* Classification et analyse ordinale des données.

2.1. Calcul d'une pondération initiale

Deux pondérations initiales de l'ensemble des sommets sont prévus conformément à l'usage de l'une des deux distances d_1 ou d_2 entre un sommet et une chaîne (cf. Chap. 9 § III.2.) ; on adoptera

$p(IX) = -n(IX)$, si la distance considérée est d_1
 et $q(IX) = \sum_{IY \leftarrow T(IX)} \{n(IX) \times \sum_{1 \leq J \leq m} |IY(J) - IX(J)|\}$, si la distance considérée est d_2 .

2.2. Calcul des r-poids et des indicateurs d'antécédence

Rappelons que le r-poids d'un sommet IX est le poids du chemin le moins "lourd" reliant l'origine IO au point IX ; $r(IX)$ est la somme de sa propre pondération initiale : $p(IX)$ (resp. $q(IX)$) si la distance utilisée est d_1 (resp. d_2) et du r-poids de son prédécesseur le moins pesant. L'indicateur d'antécédence du sommet IX sera un nombre binaire à m moments ; ceux égaux à 1 indiqueront les prédécesseurs de IX également les moins pesants. Si l'ensemble de ces derniers est de cardinal supérieur à 1, IX sera dit sommet "multiple" ; autrement IX sera dit sommet "simple".

Exemple : Pour $m = 5$, si l'indicateur d'antécédence d'un sommet donné est 01011 ; ce sommet a trois prédécesseurs optimaux les moins pesants parallèlement à la première, la seconde et la quatrième coordonnée.

2.3. Construction des échelles optimales

Les indicateurs d'antécédence permettront de mettre en évidence, de proche en proche, les échelles optimales, en partant du sommet extrême IE et en "descendant" vers le sommet origine IO. Deux cas peuvent se présenter :

- a) Tous les sommets rencontrés sont simples ; il y a une seule échelle optimale.
- b) Certains sommets rencontrés sont multiples ; il y a plusieurs échelles optimales.

Comme nous l'avons déjà signalé ci-dessus, le principe de l'algorithme que nous avons utilisé consiste à retenir dans une variable IPM l'indice synthétique du dernier sommet multiple avant l'origine de la dernière échelle trouvée ; de revenir au sommet IPM pour découvrir tous les chemins finissants possibles permettant d'achever une échelle optimale. A chaque passage l'indicateur d'antécédence du dernier sommet multiple se "simplifie" : un des moments égaux à 1 devient égale à 0 ; jusqu'à ce que cet indicateur ne comporte plus qu'un seul moment égal à 1 ; ou alors, au passage suivant IPM devient l'indice synthétique d'un sommet, au dessus du dernier exploité. Mais si, en redescendant, on ne rencontre pas de sommet multiple, $IPM = 0$ et la recherche est achevée. Comme on le verra de façon plus précise au paragraphe 4, à l'aide notamment d'un exemple ; c'est lorsqu'on rencontre plus d'un sommet multiple en descendant du sommet extrême, que notre algorithme a besoin d'être remanié pour une recherche exhaustive.

3 - PARAMETRES ET DONNEES DU PROGRAMME

Le support des données d'un traitement comprend

a) trois cartes obligatoires : la première définit le titre (format libre), la deuxième, sous un format imposé, précise la distance adoptée (d_1 ou d_2) ainsi que les options d'impression des cartes données, des q-poids et des r-poids ; la troisième carte définit, sous un format imposé, le nombre d'items et le nombre de modalités par item.

b) les cartes données qui sont un ensemble en nombre variable ; à chaque patron observé (de fréquence non nulle) correspond une carte donnée dont les quatre premières colonnes définissent la zone "test" qui peut judicieusement servir à numérotter la carte ; en supposant qu'on dispose de moins de 10.000 cartes, ce qui est le cas quasi-général. Le reste de la carte est de format imposé ; il contient la fréquence du patron observé et la suite des coordonnées du sommet correspondant.

On peut effectuer lors d'un même passage machine une suite de plusieurs traitements ; chacun pour un ensemble de cartes données relativement à un problème d'analyse hiérarchique donné. Pour organiser la suite des traitements il y a lieu de spécifier trois cartes obligatoires indiquant : 1 - début de traitement, 2 - fin de traitement, 3 - fin du travail.

Ordre des cartes

Les cartes suivantes dans l'ordre sont obligatoires

- 1ère carte : début de traitement ;
- 2ème carte : titre ;
- 3ème carte : distance adoptée et options d'impression ;
- 4ème carte : nombre d'items et nombre de modalités par item ;
- cartes données ;
- carte de fin de traitement, appelée 5ème carte ;
- éventuellement une suite de cartes analogues à la précédente pour un second traitement ;
- éventuellement une suite de cartes analogues à la précédente pour un troisième traitement ;
- ⋮
- dernière carte de fin de travail, appelée 6ème carte.

Dessin des cartes

- 1ère carte ; 1111 en colonnes 1 à 4 ;
- 2ème carte ; titre en format libre ;
- 3ème carte ; col. 2 : marquer 1 ou 2 selon la distance adoptée ;
col. 4 : marquer 1 si on désire l'impression des p-poids,
rien sinon ;
col. 6 : marquer 1 si on désire l'impression des q-poids,
rien sinon ;
col. 8 : marquer 1 si on désire l'impression des r-poids,
rien sinon ;

- 4ème carte ; col. 1 à 4 : nombre d'items, cadré à droite ;
col. 5 à 8 : nombre de modalités du 1er item ;
col. 9 à 12 : nombre de modalités du 2ème item ;
:
col. 41 à 44 : nombre de modalités du 10ème item ;
- carte donnée ; col. 1 à 4 : zone test ;
col. 5 à 12 : fréquence du patron observé (cadré à droite) ;
col. 13 à 16 : 1ère "coordonnée" ou code de la modalité de réponse au premier item (cadré à droite) ;
col. 17 à 20 : 2ème coordonnée ;
:
col. 49 à 52 : 10ème coordonnée ;
- carte de fin de traitement ; 2222 en colonnes 1 à 4 ;
- dernière carte de fin de travail ; 9999 en colonnes 1 à 4.

Il est important de signaler ici que la suite croissante des codes des modalités d'un même item démarre à partir de 1 ; soit (1, 2, ..., k) pour un item à k modalités.

4 - ORGANISATION ET ALGORITHMES DES PROGRAMMES

4.1. Organisation

La programmation a été effectuée de façon modulaire, par sous-programmes courts ; de la sorte, la compréhension s'en trouve facilitée et on peut procéder plus aisément à d'éventuelles modifications. Les sous-programmes utilisés sont d'importance variable ; on peut en distinguer de trois sortes

a) Les programmes de gestion et de lecture qui ne servent qu'une fois ; il s'agit du programme principal LERMAN qui fait appel aux sous-programmes de calcul et du sous-programme LECTUR qui lit et teste les cartes données.

b) Les sous-programmes de calcul : POIDD2 qui calcule les q-poids pour la distance d_2 , NOUPOI qui calcule les r-poids et les indicateurs d'antécédence et CHMOPT qui calcule et imprime les échelles.

c) Les sous-programmes de service : DEFES (DEFinition des unités d'Entrée-Sortie) précise les numéros des unités logiques d'entrée-sortie, CALPRE (CALculs PREliminaires) calcule une fois pour toutes et imprime des nombres qui interviennent constamment dans les calculs suivants et qui sont liés à la suite des nombres de modalités par item ; ICODAG permet de calculer à partir de m coordonnées d'un sommet un indice synthétique que DECODA permet de décoder pour restituer la suite des m coordonnées ; enfin CALIM (CALcul des LIMites) calcule les limites de la portée des boucles DO quand on veut parcourir tous les points à coordonnées entières d'un hyperplan transversal du parallélotope.

Détail des différents programmes mentionnés

a') LERMAN et LECTUR

a'₁) LERMAN, programme principal.

Ce programme organise l'appel des sous-programmes suivants selon l'organigramme ci-dessous.

Appel de DEFES

1 Lecture et test des cartes 1 ou 6, carte 2, carte 3.

Appel de CALPRE et tests d'erreurs

Appel de LECTUR et tests d'erreurs

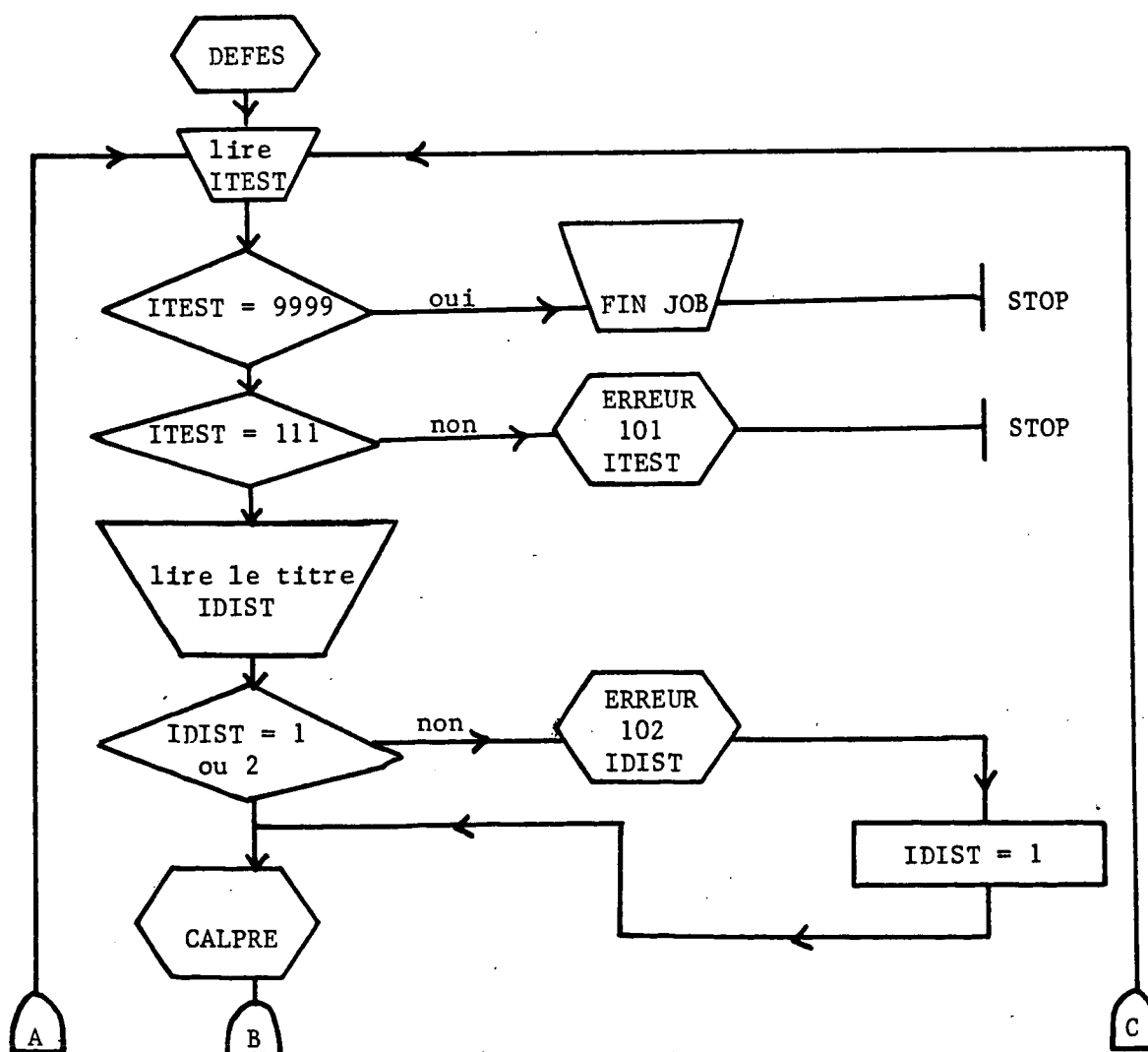
Appel de POIDD2 (cas où la distance adoptée est d_2)

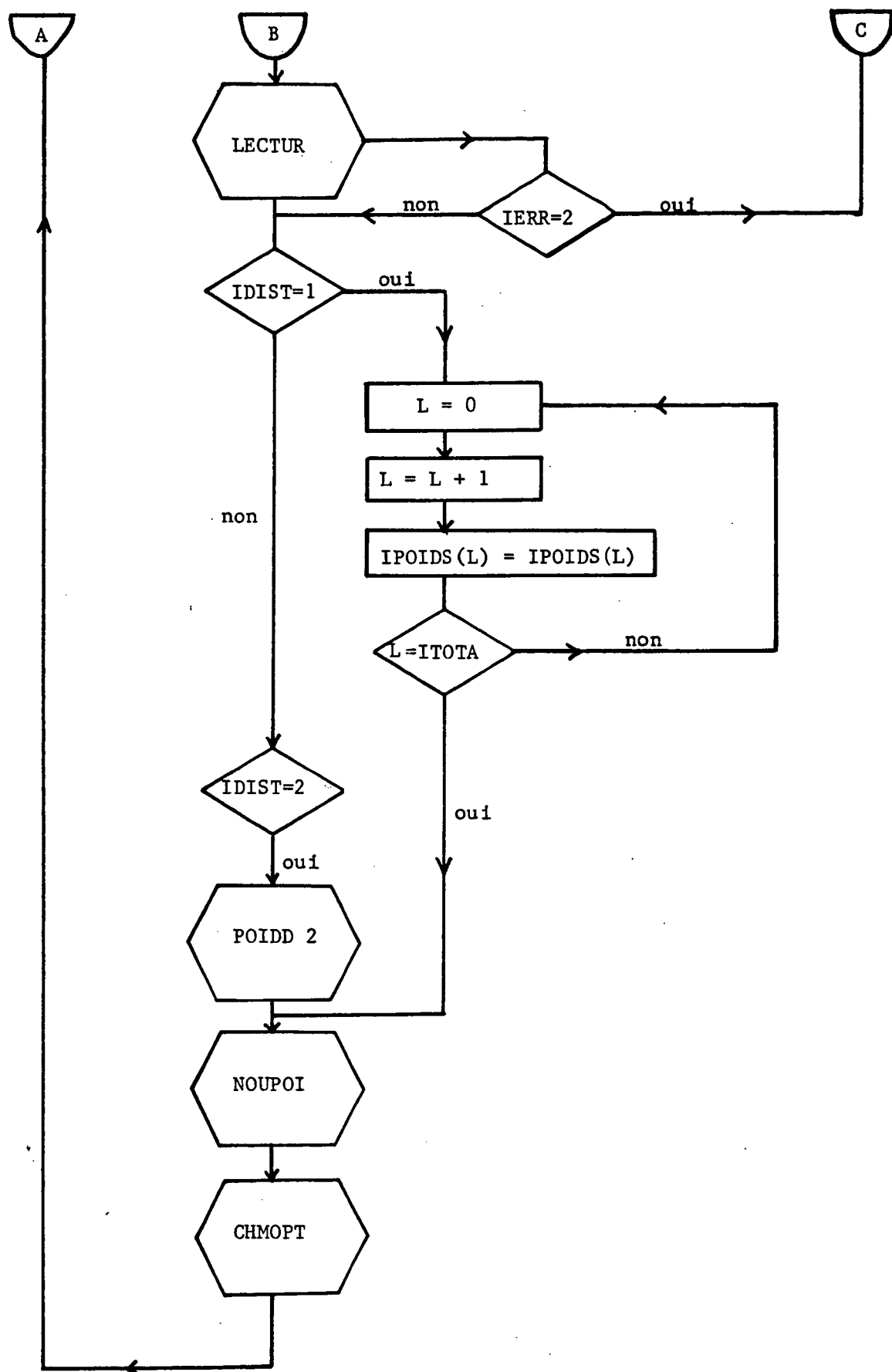
Appel de NOUPOI

Appel de CHMOPT

Retour en 1

Les diagnostics des erreurs de données possibles sont codées 1 (nombre d'items trop petit (< 2) ou trop grand (> 10) ou 2 (le treillis a plus de 20.000 sommets) ; d'autre part, ERREUR 101 est relative à la carte suivant la "5ème carte" et ERREUR 102 est relative à la 3ème carte (cf. ci-dessus).





a'₂) Sous-programme LECTUR

Ce sous-programme lit les cartes données (cf. § 3 ci-dessus) et vérifie que

1) Tous les "patrons" observés correspondent bien à des sommets du treillis ; sinon ERREUR 14 dont l'indicateur est la variable LERR.

2) Il n'existe pas plus d'une carte donnée associée à un même sommet du treillis ; sinon ERREUR 13 dont l'indicateur est la variable KERR.

3) Il n'existe pas plus de 1000 sommets du treillis chargés (i.e. correspondants à des patrons observés) ; sinon ERREUR 12 dont l'indicateur est la variable ITEST. Cette condition peut paraître restrictive ; en fait, elle est remplie dans les cas courants des enquêtes portant sur quelques milliers d'individus et ce, surtout si on tient compte de l'hypothèse d'unidimensionalité. De toute façon, on peut facilement modifier cette condition.

Un dernier indicateur d'erreur IERR pouvant valoir 1 ou 2 est calculé dans CALPRE (cf. § a'₁ ci-dessus).

Les autres variables de ce programme sont : IFR (fréquence lue de patron observé), IX(J) (J-ème coordonnée du sommet relatif), IVAL (indice synthétique d'un sommet) PLOIDS(L) (fréquence du patron observé du sommet d'indice synthétique L), NPOIDS(L) (contenu de la zone test du sommet L).

Ce programme calcule la somme des fréquences de tous les patrons observés (ISOMPA) et peut, en option, imprimer les sommets de charge non nulle.

b') POIDD2, NOUPOI et CHMOPT

b'₁) Sous programme POIDD2

Ce sous-programme qui, rappelons-le, calcule les q-poids relativement à la distance adoptée d_2 , consiste essentiellement en trois boucles :

- la plus externe est indexée par un hyperplan transversal ; elle fait varier la fonction "s" (somme des "coordonnées" d'un sommet) de $m=s$ (IO) à $S = s$ (IE). On rappelle que dans le programme, le code de la plus "petite" modalité d'un même item est 1.

- La suivante permet le calcul des "coordonnées" de chacun des points IY d'un même hyperplan transversal.

- La plus interne permet d'associer à un même sommet IY tous les points de son hyperplan pour en calculer le q-poids.

La boucle permettant d'engendrer tous les points d'un même hyperplan transversal est en fait une sorte de "boucle-itérée" comprenant (m-1) boucles imbriquées.

Dans ce programme on vérifie que les points, dont on calcule les coordonnées, appartiennent bien au treillis (sinon ERREUR 23 et 26) ; que la dimension m n'est pas inférieure à 2 (sinon ERREUR 21 et 24) ou supérieure à 10 (sinon ERREUR 22 et 25). L'une quelconque de ces erreurs entraînent l'arrêt du programme.

Les variables de ce programme sont :

IX1,IX2,...,IX10 (resp. IX(1),IX(2),...,IX(10)) : coordonnées du point IX.
 IY1,IY2,...,IY10 (resp. IY(1),IY(2),...,IY(10)) : coordonnées du point IY.
 LI2,LI3,...,LI10 : limites inférieures de portée des boucles DO sur IX(2),...,IX(10).

LS2,LS3,...,LS10 : limites supérieures de portée des boucles DO
sur IX(2),...,IX(10).
MI2,MI3,...,MI10 : limites inférieures de portée des boucles DO
sur IY(2),...,IY(10).
MS2,MS3,...,MS10 : limites supérieures de portée des boucles DO
sur IY(2),...,IY(10).
NPOIDS(L) : q-poids du point d'indice synthétique L.

Ce programme permet, en option, d'imprimer les q-poids du tous les sommets du treillis ; il y en a $\pi \bmod(j)$, où $\bmod(j)$ est le nombre de modalités de a^j .
 $1 \leq j \leq m$

b'₂) Sous-programme NOUPOI (IDIST, ISOMPA)

Ce sous-programme qui calcule des r-poids et les indicateurs d'antécédence a une structure analogue à celle du sous-programme précédent POIDD2 ; en effet, il comporte une boucle sur "s" et une "boucle-itérée" permettant de décrire tous les points de l'hyperplan transversal "s". Les variables de ce programme sont :

IDIST qui indique la distance 1 ou 2 ; ISOMPA (somme des fréquences des patrons observés) ; NPOIDS (L) (r-poids du sommet d'indice synthétique L) ; IMULT (L) (indicateur d'antécédence du sommet d'indice synthétique L) ; LVAL (indice synthétique d'un sommet précédant IX) ; il y a lieu de préciser ici qu'il n'est pas nécessaire d'utiliser la fonction ICODAG pour calculer l'indice synthétique du point (IX1, IX2, ..., IXJ-1, ..., IXM) précédant le sommet IX parallèlement à la j-ème "direction", il suffit de diminuer l'indice synthétique de IX de la quantité IPROD (J) qui vaut $\pi \bmod(j)$, non sans avoir vérifié qu'on peut diminuer la J-ème coordonnée d'une unité. IRESU est une variable intermédiaire qui contient au départ un nombre positif très grand et par la suite le r-poids du sommet précédant IX, correspondant à la plus petite valeur.

On vérifie dans ce programme que le nombre des items (i.e. la dimension) NITEM est bien supérieur à 1 (sinon ERREUR 31) et inférieur à 10 (sinon ERREUR 32) ; on vérifie que la première coordonnée est bien comprise entre ses limites (sinon ERREUR 33), pour un point courant de l'hyperplan ; qu'il en est de même des autres coordonnées (sinon ERREUR 34) et qu'enfin l'indicateur d'antécédence est inférieur à 2^m . L'une quelconque de ces erreurs provoque l'arrêt de la marche du programme.

Le programme permet, en option, l'impression des indicateurs d'antécédence et des r-poids de tous les sommets ; il y en a $\pi \bmod(j)$; d'autre
 $1 \leq j \leq m$

part il imprime la proportion $(1 - \pi)$ où π est le rapport du nombre de patrons observés situés sur une échelle optimale sur le nombre total de patrons observés (i.e. π est le rapport de la charge d'une échelle optimale sur la masse totale du treillis).

b'₃) Sous-programme CHMOPT

Ce sous-programme a pour fonction la découverte et l'impression des échelles optimales.

Nous avons déjà donné un aperçu au paragraphe 2 de l'algorithme que nous précisons au paragraphe 4.2., permettant de retrouver, à partir des indicateurs d'antécédence, les diverses échelles optimales.

Une même échelle est représentée à l'impression selon une colonne dont la i-ème ligne est la suite des coordonnées du i-ème sommet de l'échelle. Le nombre d'échelles pouvant être représenté sur une même page dépend de la dimension m ; on définit N(m) : nombre maximum d'échelles à imprimer par page. On calcule et on stocke en mémoire N(m) échelles qu'on imprime ; on recommence avec la suite des échelles restantes, jusqu'à la dernière où alors, le nombre des échelles en mémoire peut être inférieur à N(m).

Les variables servant au calcul d'une échelle sont :

N : nombre d'échelles optimales ; IX(J) : J-ème coordonnée d'un point courant ; IU : coordonnée à diminuer d'une unité ; IPM : indice synthétique du dernier sommet multiple rencontré et IPOINT : rang du sommet que l'on met en évidence.

Les variables servant à la fois au calcul et à l'impression sont :

IVPOINT(NRES, IPOINT) : il s'agit de la IPOINT-ème somme des coordonnées du sommet de la NRES-ème échelle en cours de calcul ;
 $1 \leq NRES \leq NCOLO(NITEM)$ et $1 \leq IPOINT \leq ISTO$.

Les variables servant à l'impression sont :

NCOLO(NITEM) : nombre maximum d'échelles à imprimer par page ;
 ICOMPT : nombre de paquets de NCOLO(NITEM) échelles ;
 NRES : indicateur du rang sur la page de l'échelle à imprimer
 NLARG : nombre d'échelles à imprimer sur une page, $NLARG \leq NCOLO(NITEM)$
 IDEPAR et IARRIV : numéros des échelles la plus à gauche et la plus à droite d'une même page ; enfin, ISTO est le nombre de sommets de l'échelle.

Les échelles sont imprimées numérotées.

On s'est prémuni contre les erreurs suivantes :

L'indicateur d'antécédence de tout sommet doit être ≥ 1 ; la valeur 1 correspond à un sommet simple dont le prédécesseur "optimal" est selon la première "direction" ; sinon ERREUR 41.

La coordonnée à diminuer d'une unité doit être ≥ 1 , sinon ERREUR 42.

Si la dernière échelle trouvée a un sommet multiple ($IPM \neq 0$) ; ce dernier doit avoir au moins deux moments non nuls en représentation binaire ; s'il n'en a qu'un, alors ERREUR43 et s'il n'en a aucun alors ERREUR44.

L'une quelconque de ces erreurs entraînent l'arrêt de la marche du programme.

c') DEFES, CALPRE, ICODAG, DECODA et CALIM

c'₁) DEFES

Nous avons déjà signalé le caractère très technique de ce programme où on précise les numéros des unités logiques d'entrée-sortie

c'₂) Sous-programme CALPRE (IERR)

Ce sous-programme lit la 4ème carte et fait les calculs simples suivants où apparaissent les noms des variables :

ISMAX = IA(1) + IA(2) + ... + IA(NITEM) où NITEM est le nombre des items et IA(J) est le nombre de modalités du J-ème item.

ITOTA = IA(1) * IA(2) * ... * IA(NITEM)

IPROD(L) = 1 si L=1 et IA(1) * IA(2) * ... * IA(L-1), si L > 1.

IDEUX(L) = 2^(L-1), IDEUX1 = 2^{NITEM}

IERR est un indicateur d'erreur qui vaut 1 s'il n'y a pas d'erreur et 2 s'il y en a ; les erreurs diagnostiquées sont ERREUR1 (nombre d'items <1 ou >10) et ERREUR3 (nombre de sommets > 20.000).

c'3) FUNCTION ICODAG(IX)

Le rôle de cette "FUNCTION" est de condenser la suite des m coordonnées représentant un sommet en un seul nombre appelé indice synthétique, selon la formule

$$ICODAG = 1 + \sum_{1 \leq J \leq NITEM} IPROD(J) * (IX(J) - 1)$$

c'4) Sous-programme DECODA(ISYN, IY)

Ce sous-programme réalise l'opération inverse de la précédente ; c'est-à-dire, passer de l'indice synthétique ISYN d'un sommet à la suite de ses coordonnées.

c'5) Sous-programme CALIM(L, LIMINF, LIMSUP, IS, IX)

Ce sous-programme permet de trouver tous les points d'un même hyperplan transversal caractérisé par une valeur de

$$IS = \sum_{1 \leq J \leq NITEM} IX(J) ;$$

connaissant les valeurs des coordonnées $IX(NITEM), IX(NITEM-1), \dots, IX(L+1)$; le programme permet de calculer les limites admissibles de variation de la coordonnée $IX(L)$: la limite inférieure étant définie par

$$LIMINF = \max \left(IS - \sum_{1 \leq J \leq (L-1), (L+1) \leq K \leq NITEM} IA(J) - \sum IX(K), 1 \right)$$

et celle, supérieure, par

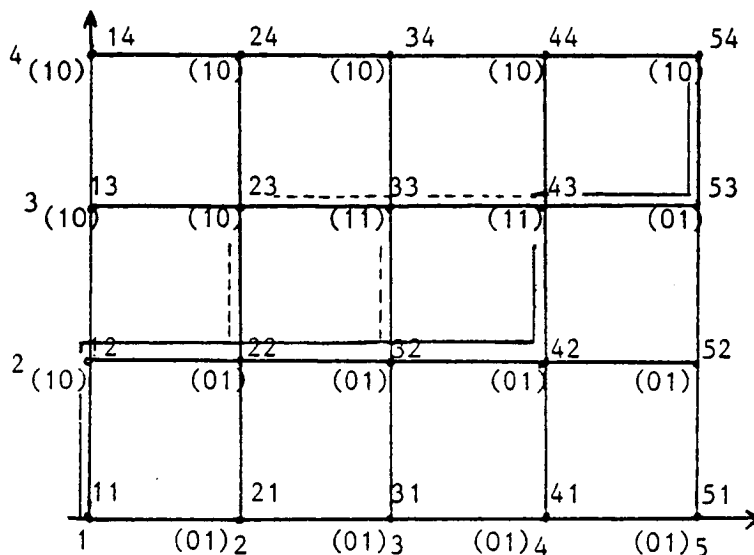
$$LIMSUP = \min \left(IS - \sum_{(L+1) \leq K \leq NITEM} IX(K) - L + 1, IA(L) \right)$$

en convenant qu'une somme étendue sur $I_1 \leq J \leq I_2$ est nulle si $I_2 < I_1$ et vaut la valeur correspondante à I_1 si $I_2 = I_1$.

4.2. Algorithmes

Dans le rapport (cf. []) dont d'ailleurs, répétons-le, nous ne faisons ici que reprendre ; C.Riso-Lévi a établi les organigrammes des algorithmes des différents programmes exprimés ci-dessus. Nous avons déjà donné ci-dessus un aperçu de ces algorithmes ; nous reviendrons seulement, à l'aide d'un exemple, sur celui qui permet d'extraire différentes échelles optimales à partir des indicateurs d'antécédence.

Soit l'exemple d'un couple d'items totaux présentant respectivement 5 et 4 modalités dont le treillis associé est représenté ci-dessous ; où nous avons attaché à chacun des sommets son indicateur d'antécédence (marqué entre parenthèses).



Nous allons indiquer le déroulement pas à pas de l'algorithme sur cet exemple où, pour des raisons de simplicité, on adoptera des notations parfois différentes de celles du programme ; ainsi, l'indice synthétique du sommet $(IX1, IX2)$ sera $L = 10 \cdot IX1 + IX2$ dont le prédécesseur optimal sera noté $PRED(L)$.

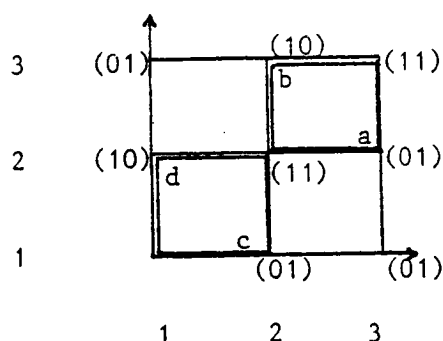
Commençons par préciser l'organigramme

- 1 - Initialisation de la recherche des échelles : $N=0$
- 2 - Initialisation de la recherche d'une échelle donnée :
 $N = N+1$ (numéro de l'échelle)
 $IPM = 0$ (on commence par le point extrême IE)
- 3 - Fin éventuelle de la recherche d'une échelle donnée
 A-t-on atteint l'origine ? ($IX=I0$)
 Si oui, aller en 4 et sinon en 5.
- 4 - Fin éventuelle de la recherche de toutes les échelles pouvant être extraites par l'algorithme.
 A-t-on épuisé tous les sommets multiples ? ($IPM=0$)
 Si oui, aller à Fin. Sinon, remplacer dans l'indicateur d'antécédence du sommet d'indice synthétique IPM , par 0, le moment égal à 1, le plus à gauche ; aller en 2.
- 5 - Traitement d'un sommet multiple
 Le sommet en question est-il multiple ?
 Sinon, aller en 6. Si oui, sauvegarder dans IPM l'indice synthétique de ce sommet.
- 6 - Passage au sommet suivant en diminuant d'une unité la coordonnée indiquée.

L'application de l'algorithme à l'exemple donne

<p> N = 0 N = 1 IPM = 0 IX = 54, IND(IX) = 10 PRED(IX) = 53 IX = 53, IND(IX) = 01 PRED(IX) = 43 IX = 43, IND(IX) = 11 IPM = 43 PRED(IX) = 42 IX = 42, IND(IX) = 01 PRED(IX) = 32 IX = 32, IND(IX) = 01 PRED(IX) = 22 IX = 22, IND(IX) = 01 PRED(IX) = 12 IX = 12, IND(IX) = 10 PRED(IX) = 11 "Fin de la recherche de la première échelle qu'on aura stockée au fur et à mesure" IND(IPM) = 01 N = N+1 IPM = 0 IX = 54, IND(IX) = 10 IX = 33, IND(IX) = 01 PRED(IX) = 23 IX = 23, IND(IX) = 10 PRED(IX) = 22 IX = 22, IND(IX) = 01 </p>	<p> PRED(IX) = 53 IX = 53, IND(IX) = 01 PRED(IX) = 43 IX = 43, IND(IX) = 01 PRED(IX) = 33 IX = 33, IND(IX) = 11 IPM = 33 PRED(IX) = 32 IX = 32, IND(IX) = 01 PRED(IX) = 22 IX = 22, IND(IX) = 01 PRED(IX) = 12 IX = 12, IND(IX) = 10 PRED(IX) = 11 "Fin de la recherche de la seconde échelle qu'on aura stockée au fur et à mesure" IND(IPM) = 01 N = N+1 IPM = 0 IX = 54, IND(IX) = 10 PRED(IX) = 53 IX = 53, IND(IX) = 01 PRED(IX) = 43 IX = 43, IND(IX) = 01 PRED(IX) = 33 PRED(IX) = 12 IX = 12, IND(IX) = 10 PRED(IX) = 11 "Fin de la recherche de la 3ème et dernière échelle" </p>
---	--

Dans ce cas, l'algorithme a permis la recherche exhaustive de toutes les échelles ; il n'en est pas de même s'il existe au moins deux sommets multiples par lesquels passent plus d'une échelle optimale. Ainsi, dans l'exemple suivant où $m = 2$, $\text{mod}(1) = 3$ et $\text{mod}(2) = 3$;



en désignant par a la suite des points $((3,3), (3,2), (2,2))$, par b celle $((3,3), (2,3), (2,2))$, par c celle $((2,2), (2,1), (1,1))$ et par d celle $((2,2), (1,2), (1,1))$; les échelles optimales sont, compte tenu des indicateurs d'antécédence marqués sur la figure, $(a,c), (a,d), (b,c)$ et (b,d) . Or le programme permet de trouver dans l'ordre $(a,c), (a,d)$ et (b,d) ; en effet, après l'extraction des deux premières échelles, l'indicateur d'antécédence du sommet $(2,2)$ est 01.

Il y a par conséquent lieu de modifier quelque peu l'algorithme en retenant tous les sommets multiples traités et ; dès que l'indicateur d'antécédence d'un sommet en cours de traitement se "simplifie" on aura soin de restituer les indicateurs d'antécédence des sommets multiples précédemment traités.

Une autre modification d'importance mineure, consiste dans la détermination des diverses échelles obtenues à partir des divers chemins "finissants" d'origine un même sommet multiple. Il suffira ici de remonter jusqu'à ce sommet multiple et non jusqu'à l'extrémité IE pour la détermination de chacune de ces échelles.

5. DESCRIPTION DES SORTIES ET QUELQUES CARACTERISTIQUES TECHNIQUES

Nous avons déjà donné un aperçu de ces sorties dans l'introduction (au paragraphe 4.1.) des différents sous-programmes ; récapitulons en précisant que les impressions peuvent se subdiviser en différentes catégories

a) Impressions relatives aux données générales du traitement :
Le titre. La distance adoptée (d_1 ou d_2). Le nombre d'items (NITEM). Le nombre de modalités par item. La somme des nombres de modalités des différents items (ISMAX). Le nombre de sommets du treillis (ITOTA)

b) Impression éventuelle des patrons observés de fréquence non nulle

c) Impression éventuelle, dans le cas de la distance d_2 , des q-poids

d) Impression éventuelle des r-poids et des indicateurs d'antécédence (en octal).

Le format d'impression pour b) c) et d) doit être reconsidéré car il est actuellement d'une ligne par sommet !

e) Impression de l'indicateur d'adéquation de l'échelle

f) Impression des échelles optimales qui sont numérotées en haut de page. Sur une même ligne figurent les sommets de même niveau de ces différentes échelles. La suite des coordonnées d'un même sommet est rangée de gauche à droite ; la j-ème étant relative au j-ème item.

La réimpression du titre signale la fin du traitement.

Ces programmes qui totalisent 500 cartes ont été écrits par C.Riso-Lévi sur C.D.C. 3600 ; ils ont par la suite été adaptés par I.Cohen sur C.D.C. 6600 où ils ont été appliqués à des données réelles. L'encombrement machine est dû :

a) au système

b) à la partie "fixe" des programmes qui comprend les programmes eux-mêmes et certaines données en COMMON, totalisant 3432 mots sur C.D.C. 3600.

c) à la partie "variable" du programme qui comprend un nombre de mots équivalent à deux fois le nombre de sommets du treillis ; cette partie peut être modulée selon la taille de la mémoire centrale dont on dispose.

Il s'agit d'un programme rapide ; signalons que le temps d'exécution des calculs (sur C.D.C. 3600), pour le traitement d'un treillis ayant 32 sommets, est de 29 secondes. Les sous-programmes dont l'exécution est la plus longue en unité centrale sont POIDD2 et NOUPOI ; si $t(s)$ est le cardinal de l'hyperplan transversal $T(s)$, on répète dans POIDD2, $\sum [t(s)]^2$ fois, une même séquence et dans NOUPOI, on répète $\sum_s t(s)$, une autre séquence.

Il peut être intéressant de pouvoir intégrer ce programme dans une chaîne de traitement dont la première partie consisterait en une classification de variables préordinales (cf. § I ci-dessus) ; une même classe faisant apparaître un ensemble d'items totaux relatif à une même "variable sous-jacente".

D'autre part, il peut également être intéressant, d'ajouter un sous-programme de calcul des q-poids pour une autre distance que celles d_1 et d_2 ; par exemple, celle d_3 (cf. chap. 9 § III.2.).

6. LISTE DES INSTRUCTIONS DU PROGRAMME

```

COMMON/UNENSO/IENT,ISBR,TITRE(20),IMP(3)
COMMON/AUXILI/NITEH,IA(10),IPRBD(10),ITBTR,IDEUX(10),IDEUX1,ISMAX
COMMON/MAXIME/IPBIDS(4500),NPBIDS(4500)
COMMON/XX/IX1,IX2,IX3,IX4,IX5,IX6,IX7,IX8,IX9,IX10
COMMON/YY/IY1,IY2,IY3,IY4,IY5,IY6,IY7,IY8,IY9,IY10
COMMON/MI/MI1,MI2,MI3,MI4,MI5,MI6,MI7,MI8,MI9,MI10
COMMON/MS/MS1,MS2,MS3,MS4,MS5,MS6,MS7,MS8,MS9,MS10
COMMON/LI/LI1,LI2,LI3,LI4,LI5,LI6,LI7,LI8,LI9,LI10
COMMON/LJ/LJ1,LJ2,LJ3,LJ4,LJ5,LJ6,LJ7,LJ8,LJ9,LJ10
CALL DEFES
10 READ(IENT,2100)ITEST
IF(ITEST,4999)110,101,110
101 WRITE(ISBR,5101)
GO TO 999
110 IF(ITEST,1111)111,120,111
111 WRITE(ISBR,5111)ITEST
GO TO 999
120 READ(IENT,2120)TITRE
WRITE(ISBR,5120)TITRE
READ(IENT,2121)IDIST,(IMP(I),I=1,3)
IF(IDIST,1)121,130,121
121 WRITE(ISBR,5121)IDIST
IDIST=1
130 WRITE(ISBR,5130)IDIST
CALL CALPRE(IERR)
CALL LECTUR(IERR,KERR,LERR,ISBMPA)
IF(IERR,2)GO TO 20
IF(KERR,1)+(LERR,1)NE.DIGB TO 10
IF(IDIST,1)130,140,130
140 DE 41 L=1,ITBTR
141 NPBIDS(L)=-IPBIDS(L)
GO TO 170
150 IF(IDIST,2)121,160,121
160 CALL PBD02
170 CALL MBUPBI(IDIST,ISBMPA)
CALL CHASPT
GO TO 10
999 STOP
2100 FORMAT(I4)
5101 FORMAT(1H1,///50X,10HFIN DE JOB,///50X,10(1H=))
5111 FORMAT(1H1,///50X,10HRENEUR 101,10X,8HITEST=,I8)
2120 FORMAT(20A4)
5120 FORMAT(1H1,///50X,5HDEBUT,///,20X,20A4)
2121 FORMAT(4I2)
5121 FORMAT(1H1,///50X,10HRENEUR 102,10X,8HIDIST=,I8)
5130 FORMAT(///50X,10HDISTANCE 0,I2)
END
SUBROUTINE DEFES
C BUT DEFINITION DES NUMEROS DE L'UNITE D'ENTREE ET DE L'UNITE DE SORTIE
COMMON/UNENSO/IENT,ISBR,TITRE(20),IMP(3)
IENT=5
ISBR=IENT+1
RETURN
END
FUNCTION ICBDAG(IX)
DIMENSION IX(10)
COMMON/AUXILI/NITEH,IA(10),IPRBD(10),ITBTR,IDEUX(10),IDEUX1,ISMAX
COMMON/UNENSO/IENT,ISBR,TITRE(20),IMP(3)
ICBDAG=1
DO 20 I=1,NITEH
ICBDAG=ICBDAG+IPRBD(I)*(IX(I)-1)
20 RETURN
END
SUBROUTINE DECBD0(I SYN,IY)
COMMON/AUXILI/NITEH,IA(10),IPRBD(10),ITBTR,IDEUX(10),IDEUX1,ISMAX
COMMON/UNENSO/IENT,ISBR,TITRE(20),IMP(3)
DIMENSION IY(10)
ITYN=ISYN-1
DO 30 I=1,NITEH
J=NITEH-1-I
IY(J)=ITYN/IPRBD(J)
IY(J)=IY(J)+1
30 IYN=ITYN,IT=IPRBD(J)
RETURN
END
SUBROUTINE CALIM(L,LIMINF,LIMSUP,IS,IX)
COMMON/AUXILI/NITEH,IA(10),IPRBD(10),ITBTR,IDEUX(10),IDEUX1,ISMAX
COMMON/UNENSO/IENT,ISBR,TITRE(20),IMP(3)
DIMENSION IX(10)
IAX=0
IIX=0
LL=L-1
IF(LL-NITEH)44,42,21
DO 20 J=LL,NITEH
IIX=IIX+IX(J)
20 LLL=LL-1
DO 30 J=1,LLL
IAX=IAX+IX(J)
LIMINF=MAX0(IS-IAX,IIX,1)
LIMSUP=MIN0(IS,IIX,L-1,IA(L))
30 RETURN
END
SUBROUTINE LECTUR(IERR,KERR,LERR,ISBMPA)
COMMON/UNENSO/IENT,ISBR,TITRE(20),IMP(3)
COMMON/AUXILI/NITEH,IA(10),IPRBD(10),ITBTR,IDEUX(10),IDEUX1,ISMAX
COMMON/MAXIME/IPBIDS(4500),NPBIDS(4500)
DIMENSION IX(10)
GO TO (300,200),IERR
200 READ(IENT,5200)ITEST
IF(ITEST,LT,1000)GO TO 200

```

```

COMMON/MI/ MI1,MI2,MI3,MI4,MI5,MI6,MI7,MI8,MI9,MI10
COMMON/MS/ MS1,MS2,MS3,MS4,MS5,MS6,MS7,MS8,MS9,MS10
COMMON/LI/ LI1,LI2,LI3,LI4,LI5,LI6,LI7,LI8,LI9,LI10
COMMON/LS/ LS1,LS2,LS3,LS4,LS5,LS6,LS7,LS8,LS9,LS10
EQUIVALENCE (MI1,MI(1)),(MS1,MS(1)),(LI1,LI(1)),(LS1,LS(1))
EQUIVALENCE (IX1,IX(1)),(IY1,IY(1))
DO 5000 I=1,10
MI(I)=0
MS(I)=0
LI(I)=0
LS(I)=0
IY(I)=0
IX(I)=0
5000 CONTINUE
NPWIDS(1)=0
ITEM=NITEM+1
DE 1000 IS=ITEM,ISMAX
GB 18 98,180,170,180,150,140,130,120,110,100,99),NITEM
98 WRITE(ISRA,9098)NITEM
GB 18 999
99 WRITE(ISRA,9099)NITEM
GB 18 999
200 CALL CALIN(10,MI10,MS10,IS,IY)
DE 300 IY10=MI10,MS10
110 CALL CALIN(9,MI9,MS9,IS,IY)
DE 890 IY9=MI9,MS9
120 CALL CALIN(8,MI8,MS8,IS,IY)
DE 380 IY8=MI8,MS8
190 CALL CALIN(7,MI7,MS7,IS,IY)
DE 570 IY7=MI7,MS7
140 CALL CALIN(6,MI6,MS6,IS,IY)
DE 860 IY6=MI6,MS6
150 CALL CALIN(5,MI5,MS5,IS,IY)
DE 850 IY5=MI5,MS5
160 CALL CALIN(4,MI4,MS4,IS,IY)
DE 840 IY4=MI4,MS4
170 CALL CALIN(3,MI3,MS3,IS,IY)
DE 830 IY3=MI3,MS3
180 CALL CALIN(2,MI2,MS2,IS,IY)
DE 820 IY2=MI2,MS2
IY1=IS
DE 260 J=2,NITEM
200 IY1=IY(J)
IF((IY1-1)=(IY1-IA(1)))210,210,201
204 WRITE(ISRA,9201)IS,(J,IY(J)),J=1,NITEM)
GB 18 999
210 IVAL=ICBDRG(IY)
MSBIDS(IVAL)=0
GB 18 (298,980,970,960,950,940,930,920,910,900,299),NITEM
298 WRITE(ISRA,9298)NITEM
GB 18 999
299 WRITE(ISRA,9299)NITEM
GB 18 999
300 CALL CALIN(10,LI10,LS10,IS,IX)
DE 800 IX10=LI10,LS10
310 CALL CALIN(9,LI9,LS9,IS,IX)
DE 790 IX9=LI9,LS9
320 CALL CALIN(8,LI8,LS8,IS,IX)
DE 780 IX8=LI8,LS8
390 CALL CALIN(7,LI7,LS7,IS,IX)
DE 770 IX7=LI7,LS7
340 CALL CALIN(6,LI6,LS6,IS,IX)
DE 760 IX6=LI6,LS6
350 CALL CALIN(5,LI5,LS5,IS,IX)
DE 750 IX5=LI5,LS5
360 CALL CALIN(4,LI4,LS4,IS,IX)
DE 740 IX4=LI4,LS4
370 CALL CALIN(3,LI3,LS3,IS,IX)
DE 730 IX3=LI3,LS3
380 CALL CALIN(2,LI2,LS2,IS,IX)
DE 720 IX2=LI2,LS2
IX1=IS
DE 400 J=2,NITEM
600 IX1=IX(J)
IF((IX1-1)=(IX1-IA(1)))410,410,401
401 WRITE(ISRA,9401)IS,(J,IX(J)),J=1,NITEM)
GB 18 999
410 ICBURA=ICBDRG(IX)
IP=IPBIDS(ICBURA)
IDEL=0
DE 420 J=1,NITEM
420 IDEL=IDEL+IABS(IX(J)-IY(J))
NPBIDS(IVAL)=NPBIDS(IVAL)+IP*IDEL
720 CONTINUE
730 CONTINUE
740 CONTINUE
750 CONTINUE
760 CONTINUE
770 CONTINUE
780 CONTINUE
790 CONTINUE
800 CONTINUE
810 CONTINUE
820 CONTINUE
830 CONTINUE
840 CONTINUE
850 CONTINUE
860 CONTINUE
870 CONTINUE
880 CONTINUE
890 CONTINUE

```

P00X009
 P00X0091
 P00X0092
 P00X0093
 P00X0094
 P00X0095
 P00X0096

 P00X0098
 P00X010
 P00X011
 P00X012
 P00X013
 P00X014
 P00X016
 P00X018
 P00X017
 P00X016
 P00X019
 P00X020
 P00X021
 P00X022
 P00X023
 P00X024
 P00X025
 P00X026
 P00X027
 P00X028
 P00X029
 P00X030
 P00X031
 P00X032
 P00X033
 P00X034
 P00X035
 P00X036
 P00X037
 P00X038
 P00X039
 P00X040
 P00X041
 P00X042
 P00X043
 P00X044
 P00X045
 P00X046
 P00X047
 P00X048
 P00X049
 P00X050
 P00X051
 P00X052
 P00X053
 P00X054
 P00X055
 P00X056
 P00X057
 P00X058
 P00X059
 P00X060
 P00X061
 P00X062
 P00X063
 P00X064
 P00X065
 P00X066
 P00X067
 P00X068
 P00X069
 P00X070
 P00X071
 P00X072
 P00X073
 P00X074
 P00X075
 P00X076
 P00X077
 P00X078
 P00X079
 P00X080
 P00X081
 P00X082
 P00X083
 P00X084
 P00X085
 P00X086
 P00X087
 P00X088
 P00X089
 P00X090
 P00X091
 P00X092
 P00X093
 P00X094
 P00X095


```

900 CONTINUE
1000 CONTINUE
IF (IMP(2)) 1010, 1020, 1010
1010 WRITE (ISBR, 4010)
      WRITE (ISBR, 4011) (I, I=1, NITEM)
      WRITE (ISBR, 4012)
      DO 1019 I=1, ITOTR
      CALL DECORR(I, IX)
      WRITE (ISBR, 4019) NPIDS(I), (IX(I), I=1, NITEM)
1019 CONTINUE
1020 CONTINUE
9201 FBRAT(1H1, 10X, 10HERREUR 25, ///10X, 3HIS=, IS, /10X, 3HIY(I2, 2H)=, I
18))
9298 FBRAT(1H1, 10X, 10HERREUR 24, ///10X, 8HNITEM=, IS)
9299 FBRAT(1H1, 10X, 10HERREUR 25, ///10X, 8HNITEM=, IS)
9401 FBRAT(1H1, 10X, 10HERREUR 26, ///10X, 3HIS=, IS, /10X, 3HIY(I2, 2H)=, IS)
9098 FBRAT(1H1, 10X, 10HERREUR 21, ///10X, 8HNITEM=, IS)
9099 FBRAT(1H1, 10X, 10HERREUR 22, ///10X, 8HNITEM=, IS)
4010 FBRAT(1H1, ///, 1X, 40HIMPRESSION DES PRIDS PSUR LA DISTANCE D2, ///)
4811 FBRAT(24X, 8HPIDS D2, 5X, 11HCS6860NNEES, 10IS)
4012 FBRAT(///)
4018 FBRAT(22K, 11D, 15X, 10IS)
      RETURN
999 STOP
END
SUBROUTINE NUPBI(IDIST, ISMPR)
  COMMON/UNEN98/IENT, ISBR, TITRE(20), IMP(9)
  COMMON/HUXICI/NITEM, IA(10), IPRED(10), ITOTR, IDEUX(10), IDEUX1, ISMAX
  DIMENSION IX(10), IMULT(4000)
  COMMON/MAXINE/IPIDS(1500), NPIDS(4500)
  COMMON/XX/IX1, IX2, IX3, IX4, IX5, IX6, IX7, IX8, IX9, IX10
  DIMENSION HI(10), HS(10), LI(10), LS(10)
  COMMON/LI/LI1, LI2, LI3, LI4, LI5, LI6, LI7, LI8, LI9, LI10
  COMMON/LS/LS1, LS2, LS3, LS4, LS5, LS6, LS7, LS8, LS9, LS10
  EQUIVALENCE (IPIDS(1), IMULT(1)), (IX1, IX(1))
  EQUIVALENCE (LI1, LI(1)), (LS1, LS(1))
  DO 5000 I=1, 10
    LI(I)=0
    LS(I)=0
    IX(I)=0
5000 CONTINUE
    IMULT(1)=0
    IITER=NITEM+1
    DO 500 IS=IITER, ISMAX
      GE TB(39, 180, 170, 160, 150, 140, 130, 120, 110, 100, 99), NITEM
98 WRITE (ISBR, 3098) NITEM
      GE TB 999
99 WRITE (ISBR, 3099) NITEM
      GE TB 999
100 CALL CALIN(10, LI10, LS10, IS, IX)
      DO 400 IX10=LI10, LS10
110 CALL CALIN(9, LI9, LS9, IS, IX)
      DO 390 IX9=LI9, LS9
120 CALL CALIN(8, LI8, LS8, IS, IX)
      DO 380 IX8=LI8, LS8
130 CALL CALIN(7, LI7, LS7, IS, IX)
      DO 370 IX7=LI7, LS7
140 CALL CALIN(6, LI6, LS6, IS, IX)
      DO 360 IX6=LI6, LS6
150 CALL CALIN(5, LI5, LS5, IS, IX)
      DO 350 IX5=LI5, LS5
160 CALL CALIN(4, LI4, LS4, IS, IX)
      DO 340 IX4=LI4, LS4
170 CALL CALIN(3, LI3, LS3, IS, IX)
      DO 330 IX3=LI3, LS3
180 CALL CALIN(2, LI2, LS2, IS, IX)
      DO 320 IX2=LI2, LS2
      IX1=IS
      DO 210 J=2, NITEM
210 IX1=IX1-IX(J)
      IF (IX1-1)=IX1-IA(1)) 220, 220, 211
211 WRITE (ISBR, 3211) IS, (J, IX(J), J=1, NITEM)
      GE TB 999
220 IVAL=ICORR(IX)
      IRESU=1000000000
      LINT=NPIDS(IVAL)
      IMULT(IVAL)=0
      DO 250 J=1, NITEM
      IF (IX(J)-1) 225, 250, 230
225 WRITE (ISBR, 3225) IS, IX
      GE TB 999
230 IVAL=IVAL-IPRED(J)
      IF (IRESU-NPIDS(LVAL)) 250, 232, 231
231 IMULT(IVAL)=IDEUX(J)
      NPIDS(LVAL)=NPIDS(LVAL)+LINT
      IRESU=NPIDS(LVAL)
      GE TB 250
232 IMULT(IVAL)=IMULT(IVAL)+IDEUX(J)
250 CONTINUE
      IF (IMULT(IVAL)-IDEUX1) 252, 251, 251
251 WRITE (ISBR, 3251) IS, IVAL, IMULT(IVAL), (J, IX(J), J=1, NITEM)
      GE TB 999
320 CONTINUE
330 CONTINUE
340 CONTINUE
350 CONTINUE
360 CONTINUE
370 CONTINUE
380 CONTINUE
390 CONTINUE
990 CONTINUE
999 CONTINUE

```

```

400 CONTINUE
500 CONTINUE
IF (IDIST-1) 520, 510, 520
510 X50MPA=IS0MPA
Z50MPA=IS0MPA.NP0IDS(IT0TA)
RAPP=Z50MPA/X50MPA
WRITE (IS0R, 3510) RAPP
520 CONTINUE
IF (IMP(3)) 610, 620, 610
610 WRITE (IS0R, 3610)
WRITE (IS0R, 3611) ((I, I=1, NITEM))
WRITE (IS0R, 3612)
DO 613 L=1, ITATA
CALL DEC0DA(L, IX)
WRITE (IS0R, 3613) IMULT(L), NP0IDS(L), ((IX(I), I=1, NITEM))
613 CONTINUE
620 CONTINUE
3098 F0RMAT (1H1, 10X, 10HERRREUR 31, /// 10X, 6HNITEM=.I6)
3099 F0RMAT (1H1, 10X, 10HERRREUR 32, /// 10X, 6HNITEM=.I6)
3211 F0RMAT (1H1, 10X, 10HERRREUR 33, /// 10X, 3HIS=.I6/(10X, 3HIX(I2, 2H)=I6))
3225 F0RMAT (1H1, 10X, 10HERRREUR 34, /// 10X, 3HIS=.I6/(10X, 3HIX(I2, 2H)=I6))
3251 F0RMAT (1H1, 10X, 10HERRREUR 35, /// 10X, 3HIS=.I6, 10X, 6HIMULT(, I6, 2H)=, NP078
1I6, /// 10X, 3HIX(I2, 2H)=I6))
3510 F0RMAT (1H1, 9X, 27HRAFFORT POUR LA DISTANCE 01, /// 50X, 79HNBRE T0TAN0UR080
1 DE PATRONS 0BSERVES-N0MBRE DE PATRONS SITES SUR UN CHEMIN 0PTI0M0UR081
2AL, /// 50X, 18H0IVISE PAR LE N0MBRE T0TAL DE PATRONS 0BSERVES, /// 40X, NP082
3F10, 51H1)
3610 F0RMAT (1H1, ///, 50X, 50HIMPRESSION DES MULTIPLICITES ET DES N0UVEAUX NP083
IP0IDS, ///)
3611 F0RMAT (7X, 19HMULTIPLICITE(0CTAL), 5X, 19HN0UVEAU PSIDS, 5X, 11H00R000N NP084
1NEES, 10I6)
3612 F0RMAT (///)
3613 F0RMAT (19X, A5, 14X, I10, 16X, 10I6)
RETURN
999 STOP
END
SUBROUTINE CH0RPT
DIMENSION NC0LB(10), IVP0IN( 0, 100), IX(10), IZ(10), IXX(10, 8)
COMMON/UNEN0B/IENT, IS0R, TITRE(20), IMP(3)
COMMON/AUXILI/NITEM, IA(10), IP00D(10), IT0TA, IDEUX(10), IDEUX1, IS0MAX
COMMON/MAXIME/IP0IDS(4500), NP0IDS(4500)
DIMENSION IMULT(4000)
EQUVALENCE(IP0IDS(1), IMULT(1))
DATA NC0LB /0, 0, 0, 5, 5, 5, 4, 4, 8, 3/
DATA (NC0LB(I), I=1, 10) /0, 0, 0, 5, 5, 5, 5, 4, 4, 8, 3/
IC0MPT=0
N=0
95 N=N+1
NRES=N-(N/NC0LB(NITEM))*NC0LB(NITEM)
IF (NRES) 101, 100, 101
100 NRES=NRES+NC0LB(NITEM)
101 DO 110 I=1, NITEM
110 IX(I)=IA(I)
IP0INT=1
IVP0IN(NRES, 1)=IT0TA
IP0=0
IS=IS0MAX+1
115 IS=IS-1
IVAL=IC0DRG(IX)
DO 120 J=1, NITEM
J=NITEM+1-I
IF (IMULT(IVAL)-IDEUX(J)) 120, 122, 123
120 CONTINUE
WRITE (IS0R, 3120) IVAL, IMULT(IVAL), N, ((IX(I), I=1, NITEM))
GO TO 999
122 IU=J
IX(IU)=IX(IU)-1
GO TO 126
123 IU=J
IX(IU)=IX(IU)-1
IP0=IVAL
126 IF (IX(IU)-1) 124, 125, 125
124 WRITE (IS0R, 3124) IS, IU, NRES, IVP0INT, IVP0IN (NRES, IVP0INT), ((IX(I), I=1,
1NITEM))
GO TO 999
125 INTER=IVP0IN(NRES, IVP0INT)
IP0INT=IP0INT+1
IVP0IN(NRES, IVP0INT)=INTER-IP00D(IU)
IF (NITEM-IS+1) 115, 130, 130
130 IF (IP0) 131, 150, 131
151 DO 154 I=1, NITEM
J=NITEM+1-I
IF (IMULT(IP0)-IDEUX(J)) 154, 152, 153
152 WRITE (IS0R, 3132)
GO TO 136
154 CONTINUE
155 WRITE (IS0R, 3135)
156 CALL DEC0DA(IP0, IX)
WRITE (IS0R, 3136) IP0, IMULT(IP0), ((IX(I), I=1, NITEM))
GO TO 999
C. FIN DE LA 00RTIE ANORMALE DERNIER POINT MULTIPLE-A MULTIPLICITE INF 2
140 IMULT(IP0)=IMULT(IP0)-IDEUX(J)
IF (NRES-NC0LB(NITEM)) 155, 141, 95
141 NLR0G=NC0LB(NITEM)
GO TO 151
150 NLR0G=NRES
151 IDEP0R=IC0MPT+NC0LB(NITEM)+1
WRITE (IS0R, 3151)
IARRIV=IDEP0R+NLR0G-1
IC0MPT=IC0MPT+1

```


C ECRITURE DE L'EN-TETE

GB TB (152,153,154,155,156,157,158,159,160),NITEM

152 WRITE(ISOA,3152)((IT,IT=IDEPAN,IARRIV)

GB TB 161

153 WRITE(ISOA,3153)((IT,IT=IDEPAN,IARRIV)

GB TB 161

154 WRITE(ISOA,3154)((IT,IT=IDEPAN,IARRIV)

GB TB 161

155 WRITE(ISOA,3155)((IT,IT=IDEPAN,IARRIV)

GB TB 161

156 WRITE(ISOA,3156)((IT,IT=IDEPAN,IARRIV)

GB TB 161

157 WRITE(ISOA,3157)((IT,IT=IDEPAN,IARRIV)

GB TB 161

158 WRITE(ISOA,3158)((IT,IT=IDEPAN,IARRIV)

GB TB 161

159 WRITE(ISOA,3159)((IT,IT=IDEPAN,IARRIV)

GB TB 161

160 WRITE(ISOA,3160)((IT,IT=IDEPAN,IARRIV)

GB TB 161

161 ISTE=ISMAX-NITEM+1

DS 200 L=1,ISTE

DS 179 K=1,NLARG

CALL DECORA(IVPIN(K,L),IZ)

DS 178 I=1,NITEM

178 IZZ(I,K)=IZ(I)

179 CONTINUE

GB TB (182,183,184,185,186,187,188,189,190),NITEM

182 WRITE(ISOA,3182)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

183 WRITE(ISOA,3183)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

184 WRITE(ISOA,3184)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

185 WRITE(ISOA,3185)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

186 WRITE(ISOA,3186)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

187 WRITE(ISOA,3187)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

188 WRITE(ISOA,3188)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

189 WRITE(ISOA,3189)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

190 WRITE(ISOA,3190)((IZZ(I,K),I=1,NITEM),K=1,NLARG)

GB TB 200

200 CONTINUE

C FIN D'IMPRESSION D'UNE SERIE DE CHEMINS OPTIMAUX

210 IF(IPH)95,211,95

211 WRITE(ISOA,3211)TITRE

RETURN

999 STOP

C3120 FORMAT(10X,10HERRREUR 41,8HIMULTI,18,2H)=,F18.5X,2HN=,I5,/,20X,3HFIN

3120 FORMAT(10X,10HERRREUR 41,8HIMULTI,18,2H)=,I12.5X,2HN=,I5,/,20X,3HFIN

IX=,10IS

3124 FORMAT(10X,10HERRREUR 42,3HIS=,I5,10X,3HIU=,I5,10X,7HIVPIN(,I5,1H,

1I5,2H)=,I5,/,20X,3HIX=,10IS)

3132 FORMAT(10X,10HERRREUR 44)

3135 FORMAT(10X,10HERRREUR 43)

3138 FORMAT(20X,8HIMULTI,18,2H)=,I12,/,20X,3HIX=,10IS)

C3138 FORMAT(20X,8HIMULTI,18,2H)=,F16,/,20X,3HIX=,10IS)

3151 FORMAT(1H1,10X,10HCHAINS OPTIMAUX////)

3152 FORMAT(8(7X,17))

3153 FORMAT(8(14X,17))

3154 FORMAT(8(14X,17))

3155 FORMAT(5(14X,17))

3156 FORMAT(5(18X,17))

3157 FORMAT(4(19X,17))

3158 FORMAT(4(22X,17))

3159 FORMAT(3(25X,17))

3160 FORMAT(3(28X,17))

3162 FORMAT(8(6X,214))

3163 FORMAT(8(8X,314))

3164 FORMAT(5(8X,414))

3165 FORMAT(5(5X,519))

3166 FORMAT(5(5X,619))

3167 FORMAT(4(5X,719))

3168 FORMAT(4(5X,819))

3169 FORMAT(3(5X,919))

3190 FORMAT(3(5X,1019))

3211 FORMAT(1H1,/,/,/,/,30X,44HFIN DE LA RECHERCHE DES CHEMINS OPTIMAUX

10E,/,/,/,20H4/1H1)

END

OPT081

OPT082

OPT083

OPT084

OPT085

OPT086

OPT087

OPT088

OPT089

OPT090

OPT091

OPT092

OPT093

OPT094

OPT095

OPT096

OPT097

OPT098

OPT099

OPT100

OPT101

OPT102

OPT103

OPT104

OPT105

OPT106

OPT107

OPT108

OPT109

OPT110

OPT111

OPT112

OPT113

OPT114

OPT115

OPT116

OPT117

OPT118

OPT119

OPT120

OPT121

OPT122

OPT123

OPT124

OPT125

OPT126

OPT127

OPT128

OPT129

OPT130

OPT131

OPT132

OPT133

OPT134

OPT135

OPT136

OPT137

OPT138

HIERR

V - PROGRAMMES SUR LA "CLASSIFICABILITE"

HCLASS, SIMUL, LOIEX, LOIAP2 (I.C. LERMAN)

Nous nous proposons au coeur de ce paragraphe de réunir l'ensemble des programmes relatifs à l'étude de la notion de "classificabilité" introduite au chapitre 3 de "Classification et analyse ordinale des données". Il s'agit de quatre programmes HCLASS, SIMUL, LOIEX et LOIAP2.

Le premier, HCLASS est le plus important, puisque c'est celui qui établit dans un cas réel, la classificabilité (ou aptitude à être classifié) d'un ensemble muni d'une préordonnance totale, sous la forme d'une distribution $D(\omega)$ dont on rappellera ci-dessous la définition.

Le deuxième, SIMUL, permet une réalisation d'un tableau d'incidence aléatoire dans le cadre d'une hypothèse d'absence de lien N_h (cf. chap. 2 de l'ouvrage précité).

Le troisième LOIEX détermine dans un cas particulier de l'h.a.l. N_h (celui où le nombre de composantes égales à 1 par vecteur logique de description, est constant) la distribution de la v.a. définie statistiquement dans HCLASS (cf. ci-dessous) et dont $D(\omega)$ représente la distribution observée.

Le quatrième programme LOIAP2 détermine une bonne approximation du résultat de LOIEX, mais pouvant être obtenue beaucoup plus rapidement.

Ces différents programmes ont été établis en 1969 sur IBM 360-75.

V.1. PROGRAMME HCLASS

1. FONCTION GENERALE DU PROGRAMME

Comme nous venons de le signaler, le but de ce programme est de calculer la distribution $D(\omega)$ qui caractérise la classificabilité d'un ensemble fini E muni d'une préordonnance totale ω et dont nous allons rapidement rappeler la définition :

Pour chaque partie à trois éléments de l'ensemble E d'objets sont associées ses trois paires d'objets distincts parmi lesquelles on distingue, relativement à une préordonnance ω établie sur E , la paire médiane et la paire supérieure. Rappelons que la Classificabilité de (E, ω) est caractérisée par la suite décroissante $D(\omega)$ des valeurs de la variable : proportion de triplets dont la paire médiane et la paire supérieure sont écartées ; chacune des valeurs étant pondérée par la proportion des paires intervenant pour séparer.

2. ALGORITHME DE CALCUL DE LA DISTRIBUTION OBSERVEE $D(\omega)$

Dans ce programme, après la lecture de la première carte définissant le nombre d'objets NB , le traitement de la préordonnance se fait séquentiellement ; chaque élément de la séquence est un triplet d'entiers $(IRAN, M, N)$ où $IRAN$ indique le rang de la paire d'objets codés M et N ($M < N$). La lecture de la préordonnance s'effectue à partir d'un fichier sur cartes à raison de 10 triplets $(IRAN, M, N)$ par carte ($FORMAT(10(I4, I2, I2))$) ; la fin du fichier sera indiquée par une suite de trois entiers sous le même $FORMAT(I4, I2, I2)$ dont le premier est négatif qu'on perforera à la suite du dernier triplet $IRAN, M, N$.

Le nombre NB est supposé ici inférieur à 75, les rangs des $NB * (NB - 1)/2$ paires différentes sont stockés au dessus de la diagonale principale d'un tableau R de dimension 75 x 75. La partie du tableau au dessous de la diagonale principale contiendra les entiers $(R(N,M), M < N)$ où $R(N,M)$ est le rang de la classe consécutive à celle de la paire $\{M,N\}$; pour cela on se sert de la variable N1M1 qui retient la dernière case (N,M) pour laquelle $R(M,N)$ a été chargé, $R(N,M)$ est au départ posé égal à 0 si avec la paire $\{M,N\}$ débute une nouvelle classe de la préordonnance et à N1M1 sinon. Au début d'une nouvelle classe, on retrouve de proche en proche les couples (N,M) de la classe précédente qu'on charge avec le rang de la nouvelle classe : en effet, N1M1 indique la dernière paire rencontrée de la classe du préordre en cours de chargement ; à l'adresse N1M1, on retrouvera le code et donc l'adresse de l'avant dernière paire rencontrée de la classe. Dans cette adresse, on indiquera l'emplacement de l'adresse de la paire qui précède et, ainsi de suite... jusqu'à rencontre de la case contenant 0. Ainsi, une adresse qui définit le code d'une paire contient elle-même le code de la paire qui précède.

Un tableau DIST à une ligne sera supposé indexé sur l'ensemble F des paires.

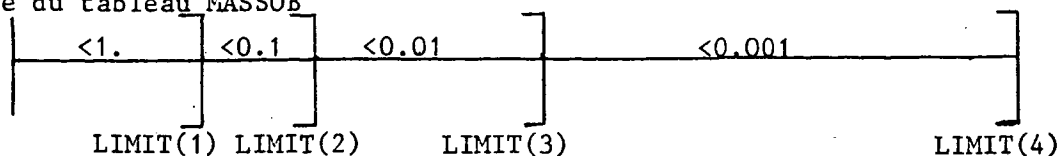
Au cours du programme, l'ensemble des triplets est passé en revue, si (I,J,K) où $I < J < K$ représente un triplet donné pour lequel on a par exemple : $R(J,K) \leq R(I,J) \leq R(I,K)$, les positions du tableau DIST correspondantes à l'intervalle ouvert]Med Sup[sont définies par l'intervalle des entiers $[R(J,I), R(I,K) - 1]$. Considérons la k-ème position de ce tableau supposé rempli.

Si $DIST(K)$ est strictement inférieur à 1., $DIST(K)$ est la proportion des triplets pour lesquelles la paire médiane et la paire supérieure enserrent la paire correspondante à la position K.

Si $DIST(K)$ est supérieur à 1., la position K correspond à un début de classe de la préordonnance, le rang correspondant étant la partie entière de $DIST(K)$ et le reste $DIST(K) - [DIST(K)]$ a la même signification que ci-dessus.

Ainsi se trouvera attachée à chacune des paires le nombre : proportion de triplets dont la paire médiane et la paire sup enserrent la paire en question ; tous les éléments d'une même classe de la préordonnance étant également affectés, un tableau MASSOB sera indexé par les numéros de la suite des classes du préordre et contiendra la valeur de cette variable proportion attachée à chacune des classes. On conjuguera au tableau MASSOB un tableau PREMP qui contiendra pour chacune des classes de la préordonnance la proportion des paires qu'elle comprend.

Pour obtenir la distribution $D(\omega)$ il nous reste à ranger de façon décroissante les valeurs du tableau MASSOB en continuant à attacher à chacune des valeurs de ce tableau, la valeur associée du tableau PREMP. Pour cela, on commence par classer les valeurs de MASSOB en quatre tranches que délimitent les pointeurs LIMIT(1), LIMIT(2), LIMIT(3) et LIMIT(4) qui contiennent les indices définissant les limites des différentes tranches rangées de gauche à droite ; ainsi LIMIT(4) est l'indice de la dernière case chargée du tableau MASSOB



On commence par constituer la tranche la plus à droite qui contiendra les valeurs de MASSOB inférieures à $\text{ALPHA} = 0.001$ puis celle consécutive à gauche qui comprendra les valeurs de MASSOB non encore classées et inférieures à 0.01 etc... comme le suggère la figure ci-dessus.

Certaines tranches peuvent être vides. LMDRN désigne l'indice de la dernière case rechargée (comptée de droite à gauche) du tableau MASSOB ; LMDR est la limite droite de balayage du tableau pour la constitution d'une nouvelle tranche.

Le programme se poursuit par le tri annoncé dans chacune des trois premières tranches (de gauche à droite) ; la dernière est délaissée pour des raisons que nous donnerons bientôt. Pour une tranche donnée, chaque passage de la boucle permettant le rangement consiste en un aller retour (de gauche à droite puis de droite à gauche) sur une partie centrale du tableau à une ligne définissant la tranche. Au bout du k -ème passage se trouvent rangées dans le bon ordre les k plus petites valeurs à l'extrême droite et les k plus grandes à l'extrême gauche. Si h est la longueur de la tranche, le nouveau balayage dans le sens gauche - droite commence à la $(k+1)$ -ème case pour atteindre la $(h-k-1)$ -ème case comparant chaque valeur avec la suivante, les inversant si elles ne sont pas dans l'ordre voulu ; de la sorte la plus petite valeur est repoussée dans la $(h-k)$ -ème case. Le balayage reprend dans le sens droite-gauche à partir de la $(h-k-1)$ -ème case pour se terminer à la $(k+2)$ -ème case plaçant dans la $(k+1)$ -ème la plus grande valeur non encore rangée. Une variable indicatrice INVRS indique en prenant la valeur 1 si inversion a été rencontrée, sinon INVRS reste égal à 0 ce qui permet de se rendre compte de la fin du rangement.

A cette étape la dimension du tableau MASSOB est souvent trop importante pour permettre la construction du graphique de la fonction en escalier dont la hauteur d'un palier donné est une valeur de la variable étudiée (chargée dans MASSOB) et l'étendue du palier le poids attaché à cette valeur (situé dans PREMP). En faisant remarquer que la statistique H (cf. Ch.3) a déjà été calculée en déterminant pour chaque triplet (I, J, K) le cardinal de l'ensemble des paires qui s'insèrent strictement entre la paire médiane et la paire sup., on se rend compte que négligeable est l'information perdue en confondant deux valeurs successives très voisines de MASSOB. Par conséquent, à chacune des tranches est associé un seuil de précision BETA, deux valeurs successives d'une même tranche MASSOB (II) et MASSOB (IJ) dont l'écart est inférieur à BETA sont condensées en la valeur la plus grande MASSOB (II), la mesure attachée à cette valeur devenant PREMP (II) + PREMP (IJ) et toute la section commençante (jusqu'à la case II) de la tranche est décalée à droite d'un cran.

Les valeurs trop petites de la dernière tranche (la plus à droite) sont condensées en la première valeur rencontrée (la plus à gauche). Cette tranche est en général la plus longue et informe peu, éviter le tri de ses valeurs est d'une grande économie.

3. LISTAGE

INTEGER * 2 R (75, 75), RR (5625)

INTEGER H/0/, SUP/0/, LIMIT(4)/4 * 0/, NAR(10), MTAB(10), NTAB(10)

REAL DIST (2775)/2775 * 0./, MASSOB(2775)/2775 * 0./, PREMP(2775)/2775*0./,

1 MSS

```

EQUIVALENCE (R(1),RR(1))
EQUIVALENCE(DIST(1),MASSOB(1))
1 FORMAT (I3)
READ(5,1) NB
I=NB*(NB-1)*(NB-2)
TRIPL=I/6.0
UNIT=1.0/TRIPL
PAIR=NB*(NB-1)/2
UNIPR=1./PAIR
NPAIR=PAIR
READ(5,2)(NAR(I),NTAB(I),I=1,10)
2 FORMAT(10(I4,I2,I2))
IDTC=1
IRAN=NAR(IDTC)
M=NTAB(IDTC)
N=NTAB(IDTC)
R(M,N)=IRAN
RAN=IRAN
DIST(IRAN)=RAN
P(N,M)=0
IPREC=IRAN
5 N1M1=N+75*(M-1)
IDTC=IDTC+1
IF(IDTC.LE.10)GO TO 17
READ(5,2)(NAR(I),NTAB(I),I=1,10)
IDTC=1
17 IF(NAR(IDTC).LT.0)GO TO 9
IRAN=NAR(IDTC)
M=NTAB(IDTC)
N=NTAB(IDTC)
R(M,N)=IRAN
RAN=IRAN
DIST(IRAN)=RAN
IF(IRAN.GT.IPREC)GO TO 7
R(N,M)=N1M1
GO TO 5
7 R(N,M)=0
IPREC=IRAN
IRET=RR(N1M1)
RR(N1M1)=IRAN
8 IF(IRET.EQ.0)GO TO 5
NMRET=IRET
IRET=RR(NMRET)
RR(NMRET)=IRAN
GO TO 8
9 CONTINUE
NBM2=NB-2
DO 100 I=1,NBM2
I1=I+1
NBM1=NB-1
DO 99 J=I1,NBM1
J1=J+1
DO 98 K=J1,NB
IF(R(I,J)-R(J,K)) 110,115,120
110 IF(R(I,K)-R(I,J)) 25,25,30
25 SUP=R(J,K)
MED=R(J,I)
GO TO 97
30 IF(R(I,K)-R(J,K)) 35,98,45
35 SUP=R(J,K)
MED=R(K,I)
GO TO 97
45 SUP=R(I,K)
MED=R(K,J)
GO TO 97
115 IF(R(I,K)-R(I,J)) 98,98,55
55 SUP=R(I,K)
MED=R(J,I)
GO TO 97
120 IF(R(I,K)-R(J,K)) 60,60,65
60 SUP=R(I,J)
MED=R(K,J)
GO TO 97
65 IF(R(I,K)-R(I,J)) 70,98,80
70 SUP=R(I,J)
MED=R(K,I)
GO TO 97
80 SUP=R(I,K)
MED=R(J,I)
97 IF(MED.EA.SUP)GO TO 98
H=H+SUP-MED
ISUP=SUP-1
DO 13 LL=MED,ISUP
13 DIST(LL)=DIST(LL)+UNIT
98 CONTINUE
99 CONTINUE
100 CONTINUE
VH=H
VH=VH/(TRIPL*PAIR)
WRITE(6,3)VH
3 FORMAT(20X,'H-CLASSIFICABILITE=',F10.8,/)
M=0
DO 150 KK=1,NPAIR
IF(DIST(KK)-1.)51,52,52
51 PREMP(M)=PREMP(M)+UNIPR

```

```

GO TO 150
52 M=M+1
   DRET=DIST(KK)
   IENT=INT(DRET)
   ENT=IENT
   MASSOB(M)=DIST(KK)-ENT
   PREMP(M)=UNIPR
150 CONTINUE
   MM1=M-1

C
C
C   FORMATION DES QUATRE TRANCHES

   ALPHA=0.001
   LIMIT(4)=MM1
   LMDR=MM1
   IDRT=M
   LMDRN=IDRT-1
   KIND=4
122 DO 4 N=2,LMDR
   IF(MASSOB(N)-ALPHA)11,24,24
   11 IDRT=IDRT-1
   LMDRN=IDRT-1
   IF(IDRT.EQ.N)GO TO 121
   IF(MASSOB(IDRT).LT.ALPHA)GO TO 11
   MSS=MASSOB(IDRT)
   MASSOB(IDRT)=MASSOB(N)
   MASSOB(N)=MSS
   PRET=PREMP(IDRT)
   PREMP(IDRT)=PREMP(N)
   PREMP(N)=PRET
24 IF(N.EQ.LMDRN)GO TO 121
4 CONTINUE

C
C
C   ENGAGEMENT DANS UNE NOUVELLE BOUCLE OU FIN DE FORMATION

121 KIND=KIND-1
   IF(LMDRN.GT.1)GO TO 222
205 LIMIT(KIND)=1
   IF(KIND.EQ.1)GO TO 200
   KIND=KIND-1
   GO TO 205
222 LIMIT(KIND)=LMDRN
   IF(KIND.EQ.1)GO TO 200
   LMDR=LMDRN
   ALPHA=ALPHA*10.
   GO TO 122

C
C
C   RANGEMENT DES VALEURS DANS CHAQUE TRANCHE

200 IND=1
   IFIN=LIMIT(IND)
   NDEP=2
151 INVRS=0
   IFIN=IFIN-1
   IF(IFIN.LT.NDEP)GO TO 300
   DO 201 IN=NDEP,IFIN
   IN1=IN-1
   IF(MASSOB(IN)-MASSOB(IN1))196,201,201
196 MSS=MASSOB(IN)
   MASSOB(IN)=MASSOB(IN1)
   MASSOB(IN1)=MSS
   INVRS=1
   PRET=PREMP(IN)
   PREMP(IN)=PREMP(IN1)
   PREMP(IN1)=PRET
201 CONTINUE
   IF(INVRS.EQ.0)GO TO 300
   INVRS=0
   NDEP=NDEP+1
   IF(IFIN.LT.NDEP)GO TO 300
   DO 202 IW=NDEP,IFIN
   IW1=IFIN-IW+NDEP
   IW1=IW-1
   IF(MASSOB(IW)-MASSOB(IW1))202,202,197
197 MSS=MASSOB(IW)
   MASSOB(IW)=MASSOB(IW1)
   MASSOB(IW1)=MSS
   INVRS=1
   PRET=PREMP(IW)
   PREMP(IW)=PREMP(IW1)
   PREMP(IW1)=PRET
202 CONTINUE
   IF(INVRS.EQ.0)GO TO 300
   GO TO 151
300 NDEP=LIMIT(IND)+1
250 IND=IND+1
   IF(IND.EQ.4)GO TO 500
   IFIN=LIMIT(IND)
   IF(IFIN.LE.NDEP)GO TO 250
   GO TO 151
500 WRITE(6,14)
14 FORMAT(1H1,TSC  REDUCTION DU TABLEAU',/)
   WRITE(6,16)
16 FORMAT(T50,'MA  '8X,'PREMP',/)
C

```

```

C   TEST DE PRECISION DANS CHACUNE DES TRANCHES
    ND=2
    IS=1
    LIMS=LIMIT(IS)
    ILIM=LIMIT(IS)-1
    BETA=0.01
    NIF=ND
    IF(LIMS.GT.NIF)GO TO 498
    IF(LIMS.LT.NIF)GO TO 533
    IF(LIMS.EQ.NIF)GO TO 544
498 DO 501 II=ND,ILIM
    IJ=II+1
    IF((MASSOB(II)-MASSOB(IJ)).GT.BETA)GO TO 501
    MASSOB(IJ)=MASSOB(II)
    PREMP(IJ)=PREMP(II)+PREMP(IJ)
C
C   DEPLACEMENT A DROITE D'UN CRAN DES SECTIONS COMMENCANTES
C
    IIM1=II-1
    DO 499 LD=NIF,IIM1
    MD=IIM1-LD+NIF
    NDP1=MD+1
    MASSOB(NDP1)=MASSOB(MD)
    PREMP(NDP1)=PREMP(MD)
499 CONTINUE
    NIF=NIF+1
501 CONTINUE
C
C   SORTIE D'UNE TRANCHE
C
12 FORMAT(T50,F6.4,8X,F8.6,/)
   WRITE(6,12)(MASSOB(I),PREMP(I),I=NIF,LIMS
   ND=LIMIT(IS)+1
   NIF=ND
555 IS=IS+1
   IF(IS.EQ.4)GO TO 600
   LIMS=LIMIT(IS)
   IF(LIMS.GT.NIF)GO TO 522
   IF(LIMS.LT.NIF)GO TO 533
   IF(LIMS.EQ.NIF)GO TO 544
544 WRITE(6,12)MASSOB(NIF),PREMP(NIF)
533 ND=LIMS+1
    NIF=ND
    GO TO 555
522 BETA=BETA/10.
    ILIM=LIMS-1
    GO TO 498
600 LIMS=LIMIT(IS)
    IF(LIMS.GT.NIF)GO TO 622
    IF(LIMS.LT.NIF)GO TO 633
    IF(LIMS.EQ.NIF)GO TO 644
622 NDP1=ND+1
    DO 601 II=NDP1,LIMS
    PREMP(ND)=PREMP(ND)+PREMP(II)
601 CONTINUE
644 WRITE(6,12)MASSOB(ND),PREMP(ND)
633 STOP
    END

```

V.2. PROGRAMME SIMUL (Simulation de l'h.a.l. N_h)

Il y a lieu de considérer ici le principe d'un programme SIMUL permettant d'engendrer une réalisation, dans l'hypothèse N, d'un tableau d'incidence aléatoire associé à un tableau de données réel.

A un vecteur de description observé de "taille" (i.e. nombre de 1 présents) fixée, sera associé un vecteur logique de même dimension et même taille mais tel que les composantes égales à 1 soient également réparties. I désignant la dimension et IT la taille du vecteur, on aura à extraire selon une loi uniforme IT entiers différents compris entre 1 et I, qui indiqueront les composantes égales à 1. A cet effet, on se sert d'un tableau ITAB à une ligne contenant au départ, dans l'ordre de gauche à droite, la suite des entiers 1, 2, ..., I. et d'une fonction permettant le choix, uniformément au "hasard", d'un entier compris entre 1 et IJ. IJ, qui au départ vaut I, décroît d'une unité après chaque désignation d'un tel entier aléatoire. Le choix de cet entier L indiquera l'adresse du tableau ITAB où se trouve le numéro d'une composante égale à 1 du vecteur de description aléatoire. La case L est vidée et rechargée avec le contenu de la case IJ du tableau ITAB ; ce faisant, on ne retrouvera pas deux fois une même composante parmi les IT à prendre.

Pour déterminer la classificabilité dans l'hypothèse N associée à un cas réel, on se servira dans la plupart des cas du programme SIMUL qui permettra quelques simulations de la matrice d'incidence ; d'où la construction de la préordonnance associée. Si la taille k du vecteur de description est sensiblement la même pour tous les objets de l'ensemble étudié (cas des questionnaires ou de certains codes descriptifs), on pourra si k reste de l'ordre de la centaine, utiliser l'un des deux programmes LOIEX ou LOIAP2 ; ce dernier étant notablement plus rapide. Si k est trop grand, une question de temps machine nous fera préférer la solution ci-dessus.

V.3. PROGRAMMES LOIEX et LOIAP2

(Classificabilité dans un cas particulier de l'hypothèse N_h)

1. CALCULS EFFECTUES

Précisons que ce cas particulier est celui où les différents objets sont considérés comme des éléments aléatoires indépendants pris dans le simplexe des parties de l'ensemble des attributs sur un même niveau K muni d'une mesure de probabilité uniformément répartie (cf. Ch. 3)

$M(m)$ (représentée dans le programme par $MASS(M)$) est la valeur de la variable attachée à la classe de la préordonnance définie par la valeur m de la similarité $s = \sum_{1 \leq i \leq I} x_i y_i$

$$M(m) = \sum_{HK} 3 \frac{\sum_{r=0}^k \binom{k-r}{p-1} \binom{k-r}{q-1} \binom{r}{1} \binom{I-(2k-r)}{k-(p+q-1)} \binom{k}{r} \binom{I-k}{k-r}}{\binom{I}{k} \binom{I}{k}}$$

où I est le cardinal de l'ensemble des attributs et HK l'ensemble :

$$\{(p,q,r) | p < m, q < m, r > m \text{ et } p+q \leq r+k, q+r \leq p+k, r+p \leq q+k\}$$

$P(m)$ (désignée ci-dessous par $PROB(M)$) est la pondération affectée à $M(m)$

$$P(m) = \binom{k}{m} \binom{I-k}{k-m} / \binom{I}{k}$$

Il s'est avéré commode d'introduire au début du programme une table de $\text{Log}(n!)$, $n=1,2,\dots,300$. Le programme de calcul de cette distribution exacte est appelé LOIEX.

Un programme analogue au précédent permet, pour des raisons de rapidité de calcul de déterminer une valeur approchée de $M(m)$,

$$M'(m) = \sum_{HK} 3 \frac{\binom{k}{p} \binom{I-k}{k-p}}{\binom{I}{k}} \times \frac{\binom{k}{r} \binom{I-k}{k-r}}{\binom{I}{k}} \times \frac{(\mu+v)^q}{q!} e^{-(\mu+v)} \quad \text{où } \mu = pr/k$$

et $v=(k-p)(k-r)/(I-k)$; $M'(m)$ sera désignée par $MASS2(M)$

Rappelons le tableau suivant permettant la comparaison des temps d'exécution de ce dernier programme, appelé LOIAP2, avec LOIEX

	LOIEX	LOIAP2
$I = 250, k=50$	40s	14s
$I = 246, k=75$	250s	51s

2 - LISTAGE - 1 - (LOI EX)

```

C      INTRODUCTION DE LA TABLE
C
      TABL(1)=0.
      DO 10 N=2,301
      X=N-1
10     TABL(N)=TABL(N-1)+ALOG(X)
C      INTRODUCTION DES DONNEES
C
      IT=250
      K=50
      WRITE(6,1)IT,K
1     FORMAT('1',T50,'T=',I4,T66,'K=',I4,////)
      KM1=K-1
      KP1=K+1
      JT=IT+1
      JTMK=JT-K
      TT=TABL(JT)
      C=ALOG(3.)+3*(TABL(KP1)+TABL(JTMK))-2*TT
      D=2*TABL(KP1)+2*TABL(JTMK)-TT
      DO 999 M=1,KM1
C      CALCUL DE MASS(M) ET PROB(M)
C
      MP1=MAX(0,M+1,2*K-IT)
      DO 100 IR=MP1,K
      DO 100 JQ=1,M
      IQ=JQ-1
      DO 100 JP=1,M
      IP=JP-1
      JR=IR+1
      IF((IP+IQ.GT. IR+K).OR.(IQ+IR.GT. IP+K).OR.(IR+IP.GT. IQ+K))GO TO 100
      LSUP=MAX(0,IP+IQ-K,IQ+IR-K,IR+IP-K)
      LINF=MIND(IT+IP+IQ+IR-3*K,IP,IQ)
      IF(LINF-LSUP)100,45,45
C      CALCUL D'UN TERME
C
45     SIG=0.
      JLSUP=LSUPX1+1
      JLINF=LINF+1
      DO 55 JL=JLSUP,JLINF
      L=JL-1
      U=TABL(JL)+TABL(JP-L)+TABL(JQ-L)+TABL(JR-L)
      V=TABL(K+JL-IP-IR)+TABL(K+JL-IQ-IR)+TABL(K+JL-IP-IQ)
      W=TABL(JT+IP+IQ+IR-3*K-L)
      REL=C-(UVW)
      EL=EXP(REL)
55     SIG=SIG+EL
      MASS(M)=MASS(M)+SIG
100    CONTINUE
      Y=TABL(MP1)+2*TABL(KP1-M)+TABL(JT+M-2*K)
      RCP=0-Y
      PROB(M)=EXP(RCP)
      WRITE(6,2)M,MASS(M),M,PROB(M)
2     FORMAT(35X,'MASS(',I3,')=',F10.4,10X,'PROB(',I3,')=',F10.4,/)
C      CALCUL DE SOMAS, SPROB ET H
C
      SOMAS=SOMAS+MASS(M)
      SPROB=SPROB+PROB(M)
      H=H+PROB(M)*MASS(M)
      IF(PROB(M)*MASS(M).LT.H*1.E-3) GO TO 66
999    CONTINUE
66     WRITE(6,3) SOMAS,SPROB,H
3     FORMAT(////,15X,SOMAS=',F10.4,10X,'SPROB=',F10.4,10X,'H=',F10.4)
      STOP
      END

```


3 - LISTAGE 2

(LOIAP2)

```

C      INTRODUCTION DE LA TABLE
C
TABL(1)=0.
DO 10 N=2,301
  I=N-1
10 TABL(N)=TABL(N-1)+ALOG(X)
C      INTRODUCTION DES DONNEES
C
IT=150
K=30
WRITE(6,1)IT,K
1 FORMAT('1',T50,'T=',I4,T66,'K=',I4,////)
KM1=K-1
KP1=K+1
JT=IT+1
JTMK=JT-K
TT=TABL(JT)
TAK=TABL(KP1)
TIMK=TABL(JTMK)
U=IT
V=K
C2=ALOG(3.)+4*(TAK+TIMK)-2*TT
D=2*TAK+2*TIMK-TT
DO 999 M=1,KM1
C      CALCUL DE MASS2(M), PROB(M) ET LEUR IMPRESSION
C
MP1=MAX0(M+1,2*K-IT)
DO 100 IR=MP1,K
DO 100 JQ=1,M
  IQ=JQ-1
DO 100 JP=1,M
  IP=JP-1
  JR=IR+1
  IF((IP+IQ.GT.IR+K).OR.(IQ+IR.GT.IP+K).OR.(IR+IP.GT.IQ+K))GO TO 100
  P=IP
  Q=IQ
  R=IR
  UMW=U-V
  PAR=P*R/V+(V-P)*(V-R)/UMV
  ATRM=C2-TABL(JP)-2*TABL(KP1-IP)-TABL(JT+IP-2*K)-TABL(JR)
  J=2*TABL(KP1-IR)-TABL(JT+IR-2*K)
  IF(PAR.EQ.0)GO TO 50
  ATRM=ATRM-TABL(JQ)-PAR+Q*ALOG(PAR)
  TRM=EXP(ATRM)
  GO TO 99
50 TRM=EXP(ATRM)
99 MASS2(M)=MASS2(M)+TRM
100 CONTINUE
  Y=TABL(MP1)+2*TABL(KP1-M)+TABL(JT+M-2*K)
  RCP=D-Y
  PROB(M)=EXP(RCP)
  WRITE(6,22)M,MASS2(M),M,PROB(M)
22 FORMAT('1',T35,'MASS2(',I3,')=',F10.4,10X,'PROB(',I3,')=',F10.4,////)
  SOMAS2=SOMAS2+MASS2(M)
  SPROB=SPROB+PROB(M)
  H2=H2+PROB(M)*MASS2(M)
  IF(PROB(M)*MASS2(M).LT.H2*1.E-3)GO TO 66
999 CONTINUE
C      SORTIE
C
66 WRITE(6,32)SOMAS2,SPROB,H2
32 FORMAT('1',T15X,SOMAS2=',F10.4,10X,'SPROB=',F10.4,10X,
)H2=',F10.4)
  STOP
  END

```

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

